Proceedings of the 16th International Conference on Educational Data Mining

Mingyu Feng, Tanja Käser & Partha Talukdar (eds).

11–14 July, 2023
Indian Institute of Science Campus
Bengaluru, India

Download copies of this and other EDM proceedings from:

International Educational Data Mining Society (IEDMS)
https://educationaldatamining.org

Proceedings of the 16th International Conference on Educational Data Mining.
Mingyu Feng, Tanja Käser & Partha Talukdar (eds).
July 11–14, 2023. Bengaluru, India.
ISBN 978-1-7336736-4-8

Preface

The Indian Institute of Science is proud to host the fully in-person sixteenth iteration of the International Conference on Educational Data Mining (EDM) during July 11-14, 2023. EDM is the annual flagship conference of the International Educational Data Mining Society.

The theme of this year's conference is "*Educational data mining for amplifying human potential.*" Not all students or seekers of knowledge receive the education necessary to help them realize their full potential, be it due to a lack of resources or lack of access to high quality teaching. The dearth in high-quality educational content, teaching aids, and methodologies, and non-availability of objective feedback on how they could become better teachers, deprive our teachers from achieving their full potential. The administrators and policy makers lack tools for making optimal decisions such as optimal class sizes, class composition, and course sequencing. All these handicap the nations, particularly the economically emergent ones, who recognize the centrality of education for their growth. EDM-2023 has striven to focus on concepts, principles, and techniques mined from educational data for *amplifying* the potential of all the stakeholders in the education system.

The spotlights of EDM-2023 include:

- Five keynote talks by outstanding researchers of eminence
- A plenary Test of Time award talk and a Banquet talk
- Five tutorials (foundational as well as advanced)
- Four thought provoking panels on contemporary themes
- Peer reviewed technical paper and poster presentations
- Doctoral students consortium
- An enchanting cultural programme.

The keynote speakers are: Sihem Amer-Yahia (CNRS, France), Ayelet Baram-Tsabari (Israel Institute of Technology, Israel), Anand Deshpande (Persistent Systems, India), Hiroaki Ogata (Kyoto University, Japan), and Jeffrey D. Ullman (Stanford University, Turing Laureate, USA). We are honoured to have them as keynote speakers. Cristina Conati (University of British Columbia, Canada) is the plenary speaker, honoured for winning the 2022 Prof. Ram Kumar EDM Test of Time Paper Award, teaming up with Saleema Amershi (Microsoft Research, USA). D.N. Prahlad (Surya Soft, India) is the banquet speaker.

The programme features five tutorials and four panel sessions. The tutorials are: (1) Core methods in EDM; (2) Introduction to neural networks and uses in EDM; (3) Learning through Wikipedia and generative AI technologies; (4) Data efficient machine learning for educational content creation; and (5) How to open science: Promoting principles and reproducibility processes within the EDM community. The panels are: (1) Turing prize worthy research problems in EDM; (2)

MOOCs: Hype or transformative force for amplifying human potential?; (3) Education in the age of generative AI; (4) Indian national education policy: EDM opportunities.

The venue is the sylvan campus of the premier research and education institution of India, the Indian Institute of Science. The host city, Bengaluru, aka Bangalore, is known variously as the Silicon Valley of India, Hi-tech industry capital of India, and the startup capital of India. It is also famous for its historical and cultural roots.

EDM 2023 received 68 submissions to the full papers track (10 pages), 36 to the short papers track (6 pages), and 15 to the poster and demo track (4 pages). The program committee accepted 18 full papers, 11 short papers, and 10 posters. The conference also provided a venue for selected papers from the Journal of Educational Data Mining to be presented to a live audience.

EDM 2023 also continued its tradition of providing opportunities for young researchers to present their work and receive feedback from their peers and senior researchers. The doctoral consortium this year features 10 such presentations.

A highlight of EDM 2023 is the tremendous geographic and gender diversity in the choice of the functional chairs and keynote speakers. It has a unique Ambassador program for young researchers and students to interact with distinguished delegates. A proud achievement of EDM 2023 is that it is providing financial support to nearly fifty first time attendees from developing nations.

We thank the sponsors of EDM 2023 for their generous support: Indian Institute of Science; Prof. Ram Kumar Memorial Foundation; Atria University; Duolingo; Accel Ventures; Infosys Technologies; Playpower Labs; Metals CMU; Carnegie Learning; Seekh; Google research; Microsoft Research India; Talent Sprint. We thank all the authors who submitted their work and the program committee members and reviewers for their expert inputs. We thank the various functional chairs for their leadership that made this conference possible. And, a big Thank You to the local arrangements committee which made this event memorable.

Mingyu Feng	WestEd	Program Chair
Tanja Käser	EPFL	Program Chair
Partha Talukdar	Google Research and Indian Institute of Science	Program Chair
Rakesh Agrawal	Data Insights Laboratories	General Chair
Y. Narahari	Indian Institute of Science	General Chair
Mykola Pechenizkiy	Eindhoven University of Technology	General Chair

July 10th, 2023
Bengaluru, India, IN

Organizing Committee

General Chairs

- Rakesh Agrawal (Data Insights Laboratories, IN)
- Y. Narahari (Indian Institute of Science, IN)
- Mykola Pechenizkiy (Eindhoven University of Technology, NL)

Program Chairs

- Mingyu Feng (WestEd, US)
- Tanja Käser (EPFL, CH)
- Partha Talukdar (Google Research and Indian Institute of Science, IN)

Diversity and Inclusion Chairs

- Olga C Santos (UNED, ES)
- Anna N. Rafferty (Carleton College, US)
- Jie Tang (Tsinghua University, CN)

Industry Track Chairs

- Giora Alexandron (Weizmann Institute of Science, IL)
- Ramasuri Narayanam (Adobe Labs, IN)
- KP Thai (Apple Inc., US)

Poster Track Chairs

- Jina Kang (University of Illinois Urbana-Champaign, US)
- Roberto Martinez-Maldonado (Monash University, AU)
- Mirka Saarela (University of Jyväskylä, FI)

Demo Track Chairs

- Yu Lu (Beijing Normal University, CN)
- Shima Salehi (Stanford University, US)
- Dmitry Ignatov (RU)

Doctoral Consortium Chairs

- Piotr Artiemjew (University of Warnia and Mazury in Olsztyn, PL)

- Min Chi (North Carolina State University, US)
- Swaprava Nath (IIT Bombay, IN)

JEDM Track Chairs

- Agathe Merceron (University of Applied Sciences, DE)
- Andrew M. Olney (University of Memphis, US)
- Maria Mercedes T. Rodrigo (Ateneo de Manila University, PH)

Workshop Chairs

- Antonija Mitrovic (University of Canterbury, NZ)
- Shaghayegh Sherry Sahebi (University at Albany – SUNY, US)
- Adish Singla (Max Planck Institute for Software Systems, DE)

Awards Chairs

- Masaru Kitsuregawa (University of Tokyo, JP)
- Cristóbal Romero (University of Córdoba, ES)
- Marianne Winslett (University of Illinois Urbana-Champaign, US)

Scholarship Chairs

- Ryan Baker (University of Pennsylvania, US)
- Rajeev Shorey (IIT Delhi, IN)
- Kalina Yacef (University of Sydney, AU)

Publicity Chairs

- Jaimie Yejean Park (Samsung Electronics, KR)
- Rohith D Vallam (IBM Research, IN)
- Jill-Jênn Vie (Inria, FR)

Web Chairs

- Paul Salvador Inventado (California State University Fullerton, US)
- Tejus Srinivas (Surfzone Technologies, IN)

Proceedings Chairs

- Qiang Ma (Kyoto Institute of Technology, JP)
- Mirko Marras (University of Cagliari, IT)

Sponsorship Chairs

- Steve Ritter (Carnegie Learning, US)
- Shourya Roy (Flipkart, IN)
- Simon Woodhead (Eedi, UK)

Local Arrangements Chairs

- Viraj Kumar (Indian Institute of Science, IN)
- Dinkar Sitaram (Cloud Computing Innovation Council of India, IN)
- Jayalakshmi D S (Ramaiah Institute of Technology, IN)

Local Organizing Committee

- Dinakar Sitaram (PES University & India Cloud Computing Innovation Council, IN)
- Jayalakshmi D S ()M S Ramaiah Institute of Technology, IN)
- Viraj Kumar (Divecha Center for Climate Change Indian Institute of Science, IN)
- Chandrika Sridhar (COMSNETS ASSOCIATION, IN)
- Nagabhushan S.V (BMSITM, IN)
- Y Narahari Professor (Center for Brain Research Indian Institute of Science, IN)
- A. Parkavi (Ramaiah Institute of Technology, IN)
- Ganeshayya I. Shidaganti (Ramaiah Institute of Technology, IN)
- S. Padmavathi (Indian Institute of Science, IN)
- T Shankar (Indian Institute of Science, IN)
- Ravikumar S P (Indian Institute of Science, IN)
- Kartik C. Sagar (Indian Institute of Science, IN)
- Akshay Nath System (Indian Institute of Science, IN)
- Aravind S (Indian Institute of Science, IN)
- Kushael (Indian Institute of Science, IN)
- Meenakshi S (Indian Institute of Science, IN)
- Nishita (Indian Institute of Science, IN)
- Shubha (Indian Institute of Science, IN)

IEDMS Officers

Tiffany Barnes,	President	North Carolina State University, US
Anna Rafferty,	Treasurer	Carleton College, US

IEDMS Board of Directors

Ryan Baker	University of Pennsylvania, US
Mingyu Feng	WestEd, US
Neil Heffernan	Worcester Polytechnic Institute, US
Sharon Hsiao	Santa Clara University, US
Tanja Käser	EPFL, CH
Kenneth Koedinger	Carnegie Mellon University, US
Kalina Yacef	University of Sydney, AU

Area Chairs

Giora Alexandron	Weizmann Institute of Science
Gautam Biswas	Vanderbilt University
Anthony F. Botelho	University of Florida
Alex Bowers	Columbia University
Christopher Brooks	University of Michigan
Alexandra Cristea	Durham University
Carol Forsyth	Educational Testing Service
Dragan Gasevic	Monash University
Sharon Hsiao	Santa Clara University
Sébastien Lallé	Sorbonne University
Andrew Lan	University of Massachusetts Amherst
Collin Lynch	North Carolina State University
Jaclyn Ocumpaugh	University of Pennsylvania
Zach Pardos	University of California, Berkeley
Philip I. Pavlik Jr.	University of Memphis
Anna Rafferty	Carleton College
Cristobal Romero	Department of Computer Sciences and Numerical Analysis
Jonathan Rowe	North Carolina State University
Shaghayegh Sahebi	University at Albany – SUNY
Olga C. Santos	aDeNu Research Group (UNED)
Kalina Yacef	University of Sidney

Program Committee

Bita Akram	North Carolina State University
Vincent Aleven	Carnegie Mellon University

Laura Allen	University of New Hampshire
Claudia Antunes	Universidade de Lisboa
Jose Azevedo	P.PORTO / ISCAP – POLITÉCNICO DO PORTO
Roger Azevedo	University of Central Florida
Ryan Baker	University of Pennsylvania
Abhinava Barthakur	University of South Australia
Prateek Basavaraj	American Association of State Colleges and Universities
Tanmay Basu	Indian Institute of Science Education and Research Bhopal
Nathaniel Blanchard	Colorado State University
Geoffray Bonnin	Université de Lorraine – LORIA
Jesus G. Boticario	UNED
Julien Broisin	Université Toulouse 3 Paul Sabatier – IRIT
Armelle Brun	LORIA – Université de Lorraine
Paulo Carvalho	Carnegie Mellon University
Guanliang Chen	Monash University
Irene-Angelica Chounta	University of Duisburg-Essen
Linda Corrin	Deakin University
Evandro Costa	Computing Institute, Federal University of Alagoas
Carrie Demmans-Epp	University of Alberta
Michel Desmarais	Ecole Polytechnique de Montreal
Spyridon Doukakis	Ionian University
Jeremiah Folsom-Kovarik	Soar Technology, Inc.
Sabine Graf	Athabasca University
Julio Guerra	University of Pittsburgh
Ella Haig	School of Computing, University of Portsmouth
Jiangang Hao	Educational Testing Service
Neil Heffernan	Worcester Polytechnic Institute
Arto Hellas	Aalto University
Erik Hemberg	ALFA
Martin Hlosta	The Swiss Distance University of Applied Sciences
Paul Hur	University of Illinois at Urbana-Champaign
Sébastien Iksal	LIUM – Le Mans Université
Paul Salvador Inventado	California State University Fullerton
Seiji Isotani	University of Sao Paulo
Vladimir Ivančević	University of Novi Sad, Faculty of Technical Sciences
Lan Jiang	University of Illinois at Urbana-Champaign
Yang Jiang	Educational Testing Service
Jina Kang	University of Illinois Urbana-Champaign
Kenneth Koedinger	Carnegie Mellon University
Irena Koprinska	The University of Sydney
Sotiris Kotsiantis	University of Patras

Juho Leinonen	Aalto University
James Lester	North Carolina State University
Qi Liu	University of Science and Technology of China
Yu Lu	Beijing Normal University
Ivan Luković	University of Novi Sad
Aditi Mallavarapu	Digital Promise and University of Pittsburgh
Mirko Marras	University of Cagliari
Noboru Matsuda	North Carolina State University
Ahmad Mel	IDLab, Ghent University
Victor Menendez-Dominguez	Universidad Autónoma de Yucatán
Agathe Merceron	Berliner Hochschule für Technik, Univ. of Applied Sciences
Donatella Merlini	Università di Firenze
Caitlin Mills	University of Minnesota
Tsunenori Mine	Kyushu University
Adway Mitra	IIT Kharagpur
Tanja Mitrovic	University of Canterbury, Christchurch
Pedro Manuel Moreno-Marcos	Universidad Carlos III de Madrid
Bradford Mott	North Carolina State University
Matthew Myers	University of Delaware
Roger Nkambou	Université du Québec à Montréal
Teresa Ober	Educational Testing Service
Andrew Olney	University of Memphis
Tounwendyam F. Ouedraogo	Université Norbert Zongo
Shalini Pandey	University of Minnesota
Luc Paquette	University of Illinois at Urbana-Champaign
Abelardo Pardo	University of South Australia
Jasabanta Patro	IISER Bhopal
Niels Pinkwart	Humboldt-Universität zu Berlin
Paul Stefan Popescu	University of Craiova
Sasha Poquet	Technical Univesity of Munich
Thomas Price	North Carolina State University
David Pritchard	Massachusetts Institute of Technology
Ramkumar Rajendran	IIT Bombay
Steven Ritter	Carnegie Learning, Inc.
Maria Mercedes T. Rodrigo	Ateneo de Manila University
Ido Roll	Technion – Israel Institute of Technology
José Raúl Romero	University of Cordoba
Daniela Rotelli	Università di Pisa
Vasile Rus	The University of Memphis
Sreecharan Sankaranarayanan	National Institute of Technology Karnataka, Surathkal
Petra Sauer	Beuth University of Applied Sciences

Ana Serrano Mamolar	Universidad de Burgos
Yang Shi	North Carolina State University
Antonette Shibani	University of Technology, Sydney
Atsushi Shimada	Kyushu University
Stefan Slater	Teachers College
Sergey Sosnovsky	Utrecht University
Balaji Vasan Srinivasan	Adobe Research Big Data Experience Lab, Bangalore
Jun-Ming Su	National University of Tainan
Ling Tan	Australian Council for Educational Research
Khushboo Thaker	University of Pittsburgh
Stefan Trausan-Matu	University Politehnica of Bucharest
Anouschka van Leeuwen	Utrecht University
Oswaldo Velez-Langs	Universidad de Cordoba
Rémi Venant	Le Mans Université – LIUM
Tuyet-Trinh Vu	SOICT-HUST
Shuai Wang	Shanghai Jiaotong University
Stephan Weibelzahl	Private University of Applied Sciences Göttingen
Jacob Whitehill	Worcester Polytechnic Institute
Beverly Park Woolf	University of Massachusetts
Amelia Zafra Gómez	Department of Computer Sciences and Numerical Analysis
Diego Zapata-Rivera	Educational Testing Service
Yingbin Zhang	University of Illinois at Urbana-Champaign
Wenbin Zhang	Michigan Technological University
Jia Zhu	Florida International University
Amal Zouaq	Ecole Polytechnique de Montréal

Sponsors

Diamond

METALS
Masters of Educational Technology
& Applied Learning Sciences

Gold

Silver

Table of Contents

Frontmatter

Abstracts

Long Papers

Short Papers

Posters

Demonstrations

Doctoral Consortium

Tutorials

Keynotes

The Gradiance Automated Homework System

Jeffrey D. Ullman, Turing Laureate, Stanford W. Ascherman Professor of Computer Science (Emeritus), Stanford University, US

We shall describe a free automated homework system and in particular the way it tries to combat cheating and its method for giving guidance as well as assessment. Central to this effort is the idea of a "root question," which is a way of phrasing multiple-choice questions in a way that enables students with incorrect answers to be given advice and then take the same homework again, without eventually discovering the correct answers by process of elimination.

Towards AI-Powered Data-Informed Education

Sihem Amer-Yahia, CNRS Research Director, FR

The Covid-19 health crisis has seen an increase in the use of digital work platforms from video-conferencing systems to MOOC-type educational platforms and crowdsourcing and freelancing marketplaces. These levers for sharing knowledge and learning constitute the premises of the future of work. Educational technologies coupled with AI hold the promise of helping learners and teachers. However, they are still limited in terms of social interactions, user experience and learning opportunities. I will describe research at the intersection of data-informed recommendations and education theory and conclude with ethical considerations in building educational platforms.

LEAF: Learning and Evidence Analytics Framework in Japan: Connecting Researchers, Practitioners and Policy-makers

Hiroaki Ogata, Professor at Kyoto University, JP

The LEAF system is a Learning and Evidence Analytics infrastructure that supports the collection, analysis, and utilization of learning logs. LEAF system consists of a Learning Management System (LMS), an eBook reader (BookRoll), Learning Record Store (LRS), and a Learning Analytics tool (Log Palette). BookRoll works as a behavior sensor and records student log data. Log Palette analyzes and visualizes the log data obtained from BookRoll and LMS. The log data can be further used for interactive lectures, reflection, recommendations, and class improvement. LEAF system has been used in over 120 educational institutions, from elementary to higher education, within eight countries and regions. Our goal is to scientifically analyze those data, support teachers and students, and transform from "education and learning based on their experiences" into "education based on data and evidence." This talk will introduce: (1) research for supporting data-and-evidence informed education, (2) practices of data-informed education with LEAF in K12 schools and universities, and (3) policies for educational data utilization in Japan.

Challenges and Opportunities in Higher Education

Anand Deshpande, Founder and Chairman, Persistent Systems, IN

As a practitioner and recruiter of college graduates, I will share perspectives of the changes in job market and how students and colleges can explore new ways to thrive in the ever changing world.

Communicating science for amplifying human potential in a post-truth era

Ayelet Baram-Tsabari, Professor at Israel Institute of Technology, IL

Science is a communication-driven endeavor – without it, we cannot build on previous research, collaborate with practitioners, or convince policymakers and stakeholders, such as parents and students, to use the resulting technology or its outcomes. In this talk, we'll explore what science communication is and why it's crucial? What do people know, and how is that related to what they do? How do we decide who to believe in? How is our worldview, the things we love and value, related to what we know? Do people need to know what they are talking about to form an opinion? And more specifically, what do people know about AI, and how can we communicate the results of AI research to diverse audiences? Finally, we will discuss what can be done so that people can make informed decisions about scientific issues. To put it more practically: what works and what doesn't when it comes to communicating science to diverse audiences?

The Prof. Ram Kumar Educational Data Mining Test of Time Award: Combining unsupervised and supervised classification to build user models for exploratory learning environments (ELE)

Cristina Conati, Professor at University of British Columbia, CA

In this talk, I will present the approach we proposed in the paper recipient of the "The Prof. Ram Kumar Educational Data Mining Test of Time Award" for building data-driven user models that can drive real-time support to students interacting with exploratory learning environments (ELEs). I will summarize the results we obtained in the past 12 years in applying extensions of this approach to a variety of ELEs, moving to discussing lessons learned and opportunities for future research.

JEDM Presentations

Using Demographic Data as Predictor Variables: a Questionable Choice

Ryan S. Baker	University of Pennsylvania
Lief Esbenshade	Google
Jonathan Vitale	Google
Shamya Karumbaiah	University of Wisconsin

Predictive analytics methods in education are seeing widespread use and are producing increasingly accurate predictions of students' outcomes. With the increased use of predictive analytics comes increasing concern about fairness for specific subgroups of the population. One approach that has been proposed to increase fairness is using demographic variables directly in models, as predictors. In this paper we explore issues of fairness in the use of demographic variables as predictors of long-term student outcomes, studying the arguments for and against this practice in the contexts where this literature has been published. We analyze arguments for the inclusion of demographic variables, specifically claims that this approach improves model performance and charges that excluding such variables amounts to a form of 'color-blind' racism. We also consider arguments against including demographic variables as predictors, including reduced actionability of predictions, risk of reinforcing bias, and limits of categorization. We then discuss how contextual factors of predictive models should influence case-specific decisions for the inclusion or exclusion of demographic variables and discuss the role of proxy variables. We conclude that, on balance, there are greater benefits to fairness if demographic variables are used to validate fairness rather than as predictors within models.

Using Auxiliary Data to Boost Precision in the Analysis of A/B Tests on an Online Educational Platform: New Data and New Results

Adam C. Sales	Worcester Polytechnic Institute
Ethan B. Prihar	Worcester Polytechnic Institute
Johann A. Gagnon-Bartsch	University of Michigan
Neil T. Heffernan	Worcester Polytechnic Institute

Randomized A/B tests within online learning platforms represent an exciting direction in learning sciences. With minimal assumptions, they allow causal effect estimation without confounding bias and exact statistical inference even in small samples. However, often experimental samples and/or treatment effects are small, A/B tests are underpowered, and effect estimates are overly imprecise. Recent methodological advances have shown that power and statistical precision can be substantially boosted by coupling design-based causal estimation to machine-learning models of rich log data from historical users who were not in the experiment. Estimates using these techniques remain unbiased and inference remains exact without any additional assumptions. This paper reviews those methods and applies them to a new dataset including over 250 randomized

A/B comparisons conducted within ASSISTments, an online learning platform. We compare results across experiments using four novel deep-learning models of auxiliary data and show that incorporating auxiliary data into causal estimates is roughly equivalent to increasing the sample size by 20% on average, or as much as 50-80% in some cases, relative to t-tests, and by about 10% on average, or as much as 30-50%, compared to cutting-edge machine learning unbiased estimates that use only data from the experiments. We show that the gains can be even larger for estimating subgroup effects, hold even when the remnant is unrepresentative of the A/B test sample, and extend to post-stratification population effects estimators.

Best Paper AIED 2022 Presentation

CurriculumTutor: An Adaptive Algorithm for Mastering a Curriculum

K. M. Shabana Indian Institute of Technology Palakkad
Chandrashekar Lakshminarayanan Indian Institute of Technology Madras
Jude K. Anil Indian Institute of Technology Palakkad

An important problem in an intelligent tutoring system (ITS) is that of adaptive sequencing of learning activities in a personalised manner so as to improve learning gains. In this paper, we consider intelligent tutoring in the learning by doing (LbD) setting, wherein the concepts to be learned along with their inter-dependencies are available as a curriculum graph, and a given concept is learned by performing an activity related to that concept (such as solving/answering a problem/question). For this setting, recent works have proposed algorithms based on multi-armed bandits (MAB), where activities are adaptively sequenced using the student response to those activities as a direct feedback. In this paper, we propose CurriculumTutor, a novel technique that combines a MAB algorithm and a change point detection algorithm for the problem of adaptive activity sequencing. Our algorithm improves upon prior MAB algorithms for the LbD setting by (i) providing better learning gains, and (ii) reducing hyper-parameters thereby improving personalisation. We show that our tutoring algorithm significantly outperforms prior approaches in the benchmark domain of two operand addition up to a maximum of four digits.

A Data Mining Approach for Detecting Collusion in Unproctored Online Exams

Janine Langerbein
University of Duisburg-Essen
Essen, Germany
janine.langerbein@vwl.uni-due.de

Dr. Till Massing
University of Duisburg-Essen
Essen, Germany
till.massing@vwl.uni-due.de

Jens Klenke
University of Duisburg-Essen
45117 Essen, Germany
jens.klenke@vwl.uni-due.de

Michael Striewe
University of Duisburg-Essen
Essen, Germany
michael.striewe@uni-due.de

Michael Goedicke
University of Duisburg-Essen
Essen, Germany
michael.goedicke@s3.uni-due.de

Christoph Hanck
University of Duisburg-Essen
Essen, Germany
christoph.hanck@vwl.uni-due.de

ABSTRACT

Due to the precautionary measures during the COVID-19 pandemic many universities offered unproctored take-home exams. We propose methods to detect potential collusion between students and apply our approach on event log data from take-home exams during the pandemic. We find groups of students with suspiciously similar exams. In addition, we compare our findings to a proctored comparison group. By this, we establish a rule of thumb for evaluating which cases are "outstandingly similar", i.e., suspicious cases.

Keywords

collusion detection, unproctored online exams, clustering algorithms

1. INTRODUCTION

During the COVID-19 pandemic many universities, e.g., in Germany, were forced to switch to online classes. Moreover, most final exams were held online. In pre-pandemic times, computer-based final exams have already proven their worth, but with the difference that they were proctored in the classroom. During the pandemic this was mostly unfeasible and students had to take the exam from a location of their choice.

There exists a wide range of supervisory measures for take-home exams. E.g., one could use a video conference software to monitor students. At many universities, however, this is legally prohibited due to data protection regulations. The exams are therefore conducted as open-book exams, i.e., students are allowed to use notes or textbooks. Yet, students must not cooperate with each other. Any form of cooperation or collusion is regarded as attempted cheating.

To our knowledge, it exists no universally-applicable method for proctoring take-home exams. It is therefore hardly feasible to stop students from illegally working together. However, one can attempt to identify colluding students post-exam. The attempt alone could have a deterring effect on students. Research in this area, however, is scarce. [3] present a method for comparing exam event logs to detect collusion. They use a simple distance measure for time series, i.e., the event logs of two different students, to quantify the similarity of these student's exams. Building on this, we propose an alternative distance measure, as well as the use of hierarchical clustering algorithms, to detect groups of potentially colluding students. We find that our method succeeds in finding groups of students with near identical exams. Furthermore, we present an approach to categorise student groups as "outstandingly similar", by providing a proctored comparison group.

The remainder of this paper is organised as follows: Section 2 provides a brief overview of related work. Section 3.1 describes the available data. Section 3.2 presents our method, including the calculation of the distance matrices. Section 4 discusses the empirical results. Section 5 concludes.

2. RELATED WORK

Due to the limited relevance of unproctored exams at universities before the pandemic, there exists little research about this topic. Recent work from [3] presents a method for analysing exam event logs for the detection of collusion in unproctored exams. They visually compare the event logs of pairs of students and quantify these by calculating a distance measure. They find some suspicious pairs of students with very similar event logs. Still, the authors remark that these findings might be purely coincidental. We enhance their approach by including a comparison group for drawing the line between "normal degree of similarity" and "outstandingly similar".

In other contexts, collusion in exams has been a relatively well studied topic. [9, 14] quantify the similarity of pro-

J. Langebein, D. T. Massing, J. Klenke, M. Striewe, M. Goedicke, C. Hanck, and N. Reckmann. A data mining approach for detecting collusion in unproctored online exams. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 6–16, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115649

gramming exams. For this, they calculate distance measures based on student's keyboard patterns. [13] further provide an overview of relevant work in educational data mining in programming exams. Complementary, our work does not focus on keyboard patterns in programming but on the submissions of answers and achieved points in introductory statistics classes. Thus, our calculation of distance measures follows a different approach.

Furthermore, a major body of related literature focuses on a different methodology. E.g., [1, 10, 16, 18, 21] use surveys or interviews with students to collect data. Due to issues inherent to surveys and interviews, like nonresponse or incorrect responses, there is little knowledge on student collusion based on actual student behaviour. We attempt to bridge this gap by directly using student's exam data.

Generally, there exists a wide range of proctoring options during take-home exams. [6, 12] introduce and compare some of these options. A supervisor could, for example, use video conference software to observe students during the exam. This provides conditions similar to those at classroom exams and thus prevents students from colluding. Such actions have two drawbacks: First, [4] argue that proctoring take-home exams is relatively costly, so that the costs exceed the potential benefits. Second, as mentioned before, most proctoring options are strictly illegal in some countries, e.g., Germany.

On the other hand, e.g., [17] advise against unsupervised online exams. They argue that, logically, with no supvervision there is no way to prevent students from colluding during the exam. To date, there are only few studies examining the impact of unattended online examinations on the integrity of students, see, e.g., [15]. [7] use a regression model that predicts final exam scores to detect collusion in unproctored online exams. Their findings suggest that collusion took place when the final exam was not proctored. [11] compares the Grade Point Average (GPA) of students who wrote a proctored exam and students who wrote an unproctored exam. There was no evidence of a significant difference in the mean GPA between the two groups, which, however, does not establish that the students did not collaborate illegally. [5] also compare the GPA in a proctored vs. an unproctored online exam and use a regression analysis to measure student collusion. The data used in these studies were final exam scores or the GPA. None of them uses data collected during the exam.

3. METHODOLOGY

In the following we give a brief overview about the data used in our analysis. We further describe our approach to build a suitable distance metric.

3.1 Data

The data we use stems from the introductory statistics course "Descriptive Statistics" at the Faculty of Business Administration and Economics at the University Duisburg-Essen, Germany.[1] The exam of our test group was taken unproctored during the global COVID-19 pandemic in the winter term 2020/21. The exam of our comparison group took place in the winter term 2018/19, i.e., before the pandemic.[2] It was a proctored exam located in a PC-equipped classroom at the university. Both exams use the e-assessment platform JACK [20].

Table 1: This table gives summary statistics for all students considered in our empirical analysis

Year	Minutes	Points	Subtasks	Students
Comparison Group (18/19)	70	60	19	109
Test Group (20/21)	70	60	17	151

Both exams consist mainly of arithmetical problems, where students are expected to submit numerical results. Moreover, there exist some tasks where students are obliged to use the programming language R [19]. The test group also had to answer a short essay task which should contain 4-5 sentences. All but the free-text tasks are evaluated automatically by JACK. The latter is manually graded by the examiner.

During the exam, the students' activities are stored in said event logs. Hence, these contain the exact time for all inputs in all tasks. For all tasks students can change and re-submit their entries. The last submission will be evaluated. For this reason, one task can list multiple events in the event log.

In addition to the event logs we also use the points achieved per task for our analysis.

Table 1 displays the basic data for both exams. Namely, these are the duration and maximum points to achieve, as well as the number of subtasks and participants per group. The wide disparity in student participants between both exams can be explainend by a change in examination regulations. During the COVID-19 pandemic, ergo in the test group, students were allowed to fail exams without any penalties. In order to prevent this from biasing our results, we removed students who attended the exam for only a few minutes and those who achieved merely a fraction of the maximum points.[3] We also removed twelve students from the test group who reported internet problems during the exam.

[1] All personal data was pseudonymised. The chair and the authors have followed the General Data Protection Regulations (GDPR) by the EU as well as national law. Reproducible Code and toy data can be found at `https://github.com/Janine-Langerbein/EDM_Detecting_Collusion_Unproctored_Online_Exams`. For information on access to the actual dataset, please contact the Dean of the Faculty of Business Administration and Economics at the University of Duisburg-Essen (dekanat@wiwi-essen.uni-due.de).

[2] The course is jointly offered by two chairs and therefore held on a rotating basis. Hence, the exam data is only comparable every two years.

[3] We also conducted the analysis without the removal of these students, with no effect but a reduced interpretability of the following clustering algorithms.

From our perspective, the setup is reasonably comparable in both groups. Although the lecture of the comparison group was held in presence and the lecture of the test group was held online, both groups shared the same content and learning goals. Both times students were given the opportunity to ask questions during the lecture. The amount of those questions remained approximately stable. Due to the sheer size of the course, with more students attending classes than participating in the exam, direct discussions were sparse even pre-pandemic.

3.2 Model

We adopt an exploratory approach for finding clusters of students with similar event patterns and points achieved during the exam. For this, we use agglomerative, i.e., bottom-up, hierarchical clustering algorithms. The results are depicted in a dendrogram. We build on previous work by [8] and [2].

In general, clustering algorithms attempt to group N objects according to some predefined dissimilarity measure. Those objects have measurements x_{ij} for $i = 1, 2, \dots, N$, on attributes $j = 1, 2, \dots, h$. The global pairwise dissimilarity $D(x_i, x_{i'})$, with x_i being x_{ij} over all j, between two objects i and i' is defined as

$$D(x_i, x_{i'}) = \frac{1}{h} \sum_{j=1}^{h} w_j \cdot d_j(x_{ij}, x_{i'j}); \sum_{j=1}^{h} w_j = 1, \tag{3.1}$$

with $d_j(x_{ij}, x_{i'j})$ the pairwise attribute dissimilarity between values of the jth attribute and w_j the weight of the attribute. The clustering algorithm therefore takes a distance matrix as input.

In our case, the students are the objects to be clustered, with $N = 151$ students. As attributes we use the dissimilarities in the student's event patterns and the dissimilarities in their points achieved. Both need to be calculated differently, but over all subtasks. Hence, we split $d_j(x_{ij}, x_{i'j})$ into two parts.

We call the attribute dissimilarity for the points achieved $d_j^P(s_{ij}, s_{i'j})$ with w_j^P its corresponding weight. s_{ij} denotes the points achieved by student i in the jth subtask. Since there are 17 subtasks we obtain a total of $h = 34$ attributes. To receive a dissimilarity measure we calculate the absolute differences

$$d_j^P(s_{ij}, s_{i'j}) = |s_{ij} - s_{i'j}|. \tag{3.2}$$

Next, $d_j^L(v_{ij}, v_{i'j})$ describes the dissimilarities in the event patterns per subtask. To calculate these we divide the examination time into $m = 1, \dots, K$ intervals of one minute. Since both exams each took 70 minutes, we obtain $K = 70$ intervals. We count each student's answer per interval. The count is denoted with v_{ijm}.[4] To obtain a pairwise attribute dissimilarity measure for all subtasks, we calculate the Manhattan metric over all counted quantities

$$d_j^L(v_{ij}, v_{i'j}) = \sum_{m=1}^{K=70} |v_{ijm} - v_{i'jm}|. \tag{3.3}$$

The corresponding weight is denoted by w_j^L. To ensure better transparency, we provide a detailed explanation of each variable in Appendix A.

Finally, we modify (3.1) so that

$$\begin{aligned} D(s_i, s_{i'}, v_i, v_{i'}) = &\frac{1}{h} \sum_{j=1}^{h} \Big(w_j^P \cdot d_j^P(s_{ij}, s_{i'j}) \\ &+ w_j^L \cdot d_j^L(v_{ij}, v_{i'j}) \Big) \\ &\text{with} \sum_{j=1}^{h} w_j^P + w_j^L = 1. \end{aligned} \tag{3.4}$$

The attribute weights w_j control the influence of each attribute on the global object dissimilarity. If all 34 attributes are to be weighted equally, each attribute would be assigned a weight of $\frac{1}{34}$. Here, however, we weight the attributes with regard to our research question. Specifically, we observe that students submit entries more often in the case of R-tasks, viz. subtasks $6a, 6b$ and $6c$. One possible interpretation of this is that students submit their code more often to check its executability. Furthermore, task 7 demands the answer to be a short text which was corrected manually. This could lead to insufficient comparability between students due to accidental arbitrariness during correction. Based on these aspects, it appears reasonable to reduce the weight of said subtasks.

We further reduce the influence of the points achieved during the exam by decreasing their weight. This follows from the fact that prior to the exam we must define all (partially) correct answers in JACK. In doing so, it is not feasible to anticipate all types of mistakes resulting from, e.g., calculation errors made by students.[5] Students might receive no points due to careless mistakes, while still having employed a correct solution strategy. In our view, this might impede the detection of colluding students, e.g., if there exist large differences in points as one student makes more frequent careless mistakes due to the random numbers in the tasks. On this account, we assign smaller weight to the points achieved. For greater clarity, an overview with all exact final weights for all attributes can be found in Appendix B.[6]

The influence of each attribute on object dissimilarity further depends on its scale. We therefore normalise each attribute.

From these pairwise object dissimilarities, we create the distance matrices. We then apply agglomerative hierarchical clustering. This builds a hierarchy by merging the most similar pairs of students, viz. those with the lowest object dissimilarity $D(x_i, x_{i'})$, into a cluster. This is repeated $N-1$ times, until all students are merged into one single cluster. The merging process is implemented with different linkage methods. These differ in their definition of the shortest distance between clusters. Here, we use single, average and complete

[4] We consider this an enhancement of the distance measure used in [3], as it enables us to analyse exams with more than one answer per task.

[5] The tasks are randomised, i.e., there exist variations so that sharing exact results is not expedient for the students.

[6] A robustness check regarding the object weights can be found in Appendix D. In this analysis, the weights of the objects are equal. The results are basically identical.

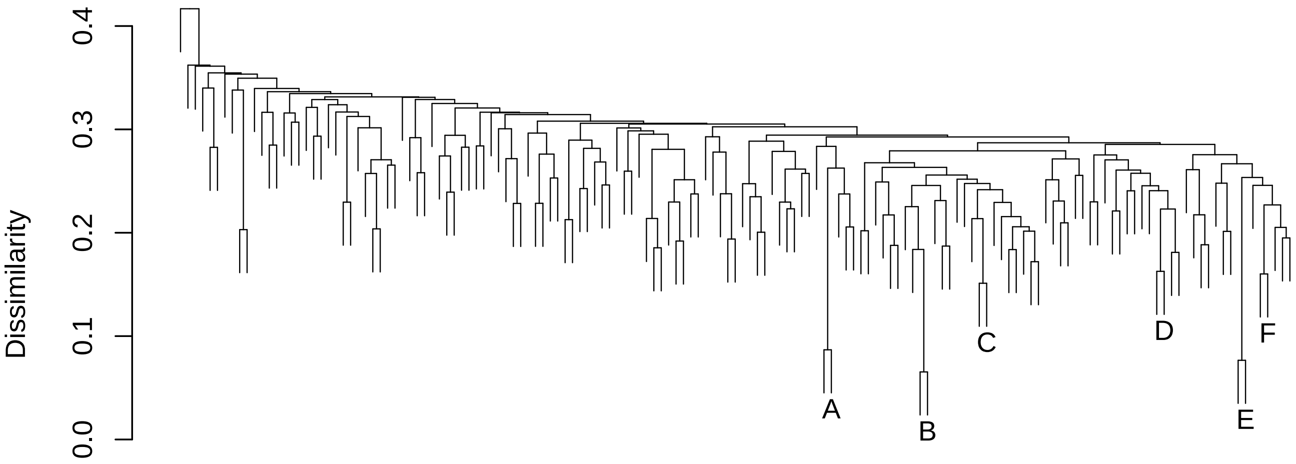


Figure 1: Dendrogram produced by average linkage clustering for the test group 2020/21. The dissimilarity of each group's node is displayed on the y-axis. Its value corresponds to the dissimilarity of the group's left and right member. A - F denotes the six lowest dissimilarity clusters. Of these especially, clusters A, B, and E are notable.

linkage. The former merges clusters with the closest minimum distance, the latter uses the closest maximum distance. The average linkage method (here: unweighted pair group method with arithmetic mean) defines the distance between any two clusters as the average distance among all pairs of objects in said clusters.

4. EMPIRICAL RESULTS

We present the results of the hierarchical clustering algorithms in a dendrogram. This provides a complete visual description of the results from the agglomerative hierarchical clustering algorithm. A dendrogram resembles a tree structure where each object is represented by one leaf. In a bottom-up approach, the objects are merged into groups one by one according to their dissimilarity. Hence, each level of the tree corresponds to one step of the clustering algorithm. The junction of a group is called a node.

There exist various hierarchical clustering algorithms. Each has a different definition of the distance between groups of observations as a function of the pairwise distances. We calculate the cophenetic correlation coefficent to assess how faithfully each algorithm represents the original structure in the data. Appendix C, Table 5 gives an overview of the cophenetic correlation coefficient for the different linkage methods for the test and comparison groups. Based on this, we deem average linkage clustering the most suitable algorithm. Figure 1 shows the corresponding dendrogram. The dissimilarity of each group's node is plotted on the vertical axis. Its value corresponds to the dissimilarity of the group's left and right member.

It is important to note that a dendrogram only gives an indication of clusters which best fit the data. It is up to the analyst to decide which are to be examined in further detail.

The dendrogram has a slightly elongated form. Still, compact clusters were produced at medium dissimilarities. This general shape is typical for the underlying algorithm, as average linkage clustering combines the long form of single linkage clustering with the smaller, tighter clusters of complete linkage clustering. Additionally, we observe three notable clusters (A, B and E) which form at a significantly lower height. Each of these three clusters consists of two students. Prima facie, this indicates the absence of collusion in larger groups.

As explained above, the hierarchical algorithm does not cluster the data itself, but imposes a structure according to the students' dissimilarities. There exist various formal methods to decide on an optimal number of clusters given this established hierarchy. Since our primary interest lies in the detection of clusters at low dissimilarities, instead of the general structure of the data, we exemplary investigate the six lowest clusters (A - F) in Figure 1.

Figure 2 shows the exact course of events for the described selection of clusters. Each scatterplot plots all answers of the students in the cluster against their time of submission. We expect students' chronology to be more similar if their cluster's node is a lower height, i.e., lower dissimilarity. We also add the points achieved on top in a barchart.

As expected, all scatterplots show some kind of similarity. In particular, clusters A, B and E bear a striking resemblance.

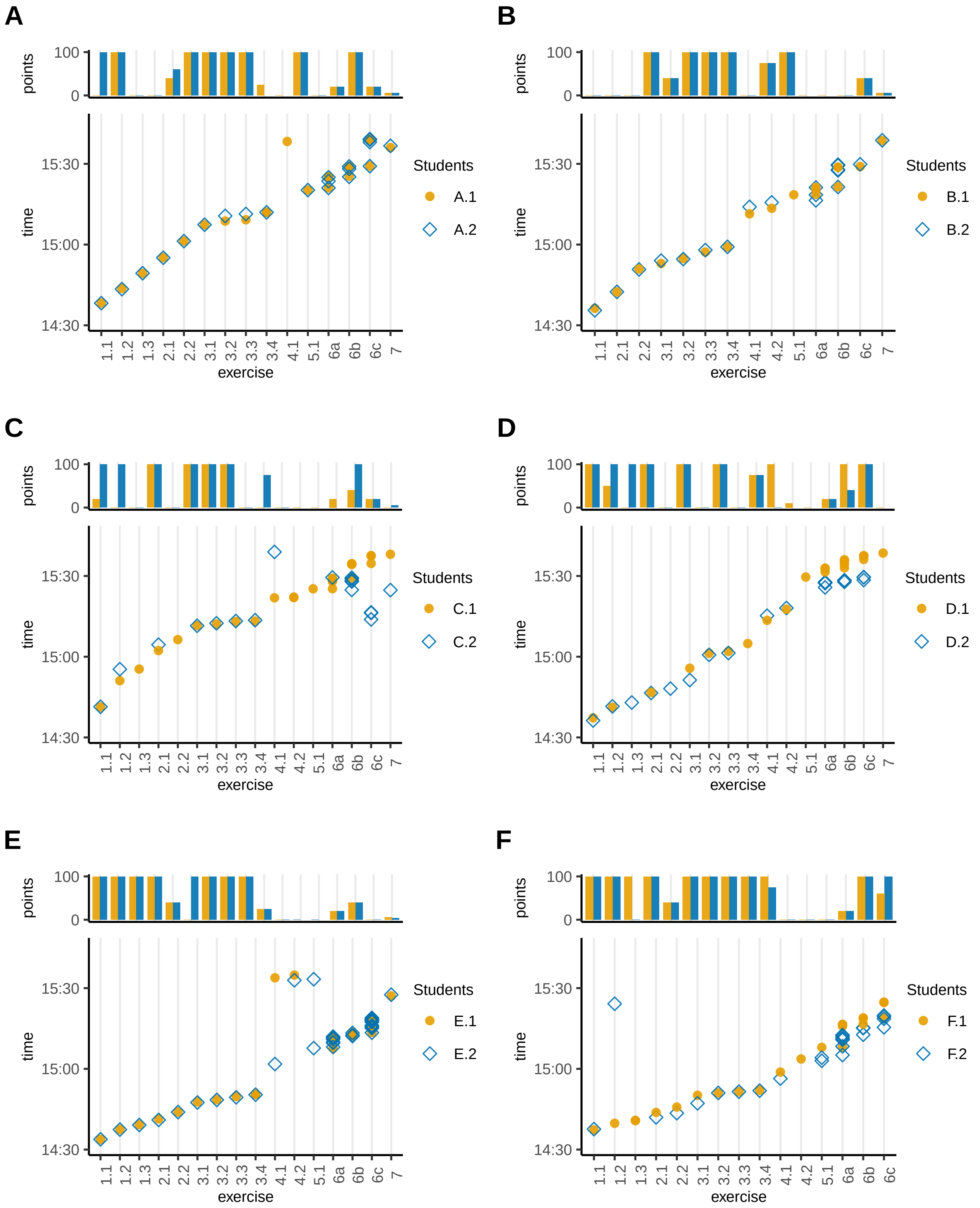


Figure 2: Comparison of the event logs and achieved points for each of the test group's (2020/21) six lowest dissimilarity clusters (A - F). The letters of the sub-figures correspond to the marked clusters in Figure 1. The number behind the letters refers to the node's left and right arm, respectively. At the bottom of each sub-figure, the sub-tasks are plotted against the clock time. Above the scatterplot, a bar chart is added to compare the points per subtask.

Their respective barchart further reveals these students to almost always achieve an equal number of points per subtask. In direct comparison, the plots of the remaining clusters C, D and F look less similar. This shows that clusters with lower node heights indeed contain more similar exams.

To assess whether these similarities are the result of collusion or coincidence, we repeat our approach on the comparison group, the final exam of the same course from two years ago. The plots were created analogously to the plots of the test group.

For the comparison group we also focus on average linkage clustering. The associated dendrogram, however, has a slightly different shape (see Figure 3). The most prominent difference, in contrast to the test group, is the absence of any visually outstanding clusters. We rather observe most nodes at a comparatively similar height.

Figure 4 shows the scatter- and barplots of the six lowest clusters in the comparison group. We find there to be significantly less similarities not only in the chronological aspect, but also in the points achieved.

The results from the comparison group support our assumption that widespread collusion over the entire exam is hardly achievable in presence. Moreover, the clear visual differences between comparison- and test group indicate that our findings in the latter might not be coincidental.

In the test group it is relatively simple to identify at least three suspicious clusters. In the less obvious cases it can be challenging to decide which clusters to investigate further, as there exists no clear rule on where to draw the line between suspicious and unsuspicious cases. We address this issue by defining a "normal degree" of similarity, which can be used as a bound to classify whether a pair of students is deemed suspicious or not. For our data of the test group, there is no indication of the existence of suspicious clusters of more than two students. Hence, we refocus on the global pairwise dissimilarity $D(x_i, x_{i'})$.

To assess the "normal degree" of similarity, we first standardise both distributions to improve their comparability. For the comparison group, we define a lower bound below which we categorise observations as extreme outliers. This bound is then used on the lower tail of the distribution of the test group. We want to identify cases in the unproctored test group which are rather extreme compared to the proctored comparison group. For this we calculate the lower bound as $Q_1 - 3 * IQR$ with Q_1 being the first quartile of the data and IQR being the interquartile range.

The boxplots in Figure 5 show the distributions of the global pairwise dissimilarity $D(x_i, x_{i'})$ of all students in the comparison- and test group. A boxplot provides a graphic overview of location and dispersion of a distribution. The eponymous box marks the upper and lower quartile of the data. Outliers are displayed by individual points.

On the left hand side of Figure 5 we observe that both distributions posess a similar shape, but a different median. The median value of the test group is significantly lower, indicating a lower average global pairwise dissimilarity in this group. Furthermore, we discover a high number of outliers in both groups, albeit at different positions in their respective distribution. In the test group, more outliers lie on the lower side of the box, with a greater distance to the main part of the distribution. We also find three observations with extremely small values on the lower tail of the test group's distribution. Unsurprisingly, these belong to the clusters A, B and E.

The right side of Figure 5 shows the normalised distributions. It is apparent that the normalised distribution of the test group still contains more outliers.

We apply the above mentioned lower bound on the test group's distribution to identify groups of students which are "outstandingly similar". Here, the before mentioned three cases (clusters A, B and E) fall below the lower bound for extreme outliers. While it is no surprise that these clusters were detected, our approach still aids us in deciding on when to stop inspecting further groups of students, as their level of similarity might as well occur in the comparison group.

To summarise, our approach offers a rule of thumb for narrowing down the number of suspicious cases. This is particularly useful if the visual distinction of cases is not clear-cut.

5. CONCLUSIONS AND DISCUSSION

During the COVID-19 pandemic many exams at universities had to be converted into unproctored take-home exams. We propose a method for detecting potentially colluding students in said exams. For this, we calculated a distance measure based on the students' event logs and their points achieved from the exam. Compared to former approaches adressing this topic, we use a distance measure which also applies if there exist multiple events per task. Subsequently, we use hierarchical clustering algorithms to detect clusters of potentially colluding students. The results show that our method is able to detect at least three clusters with near identical exams.

To decide which degree of similarity might be more than a coincidence we compare the normalised distributions of the distance measures of our test and comparison group. We find pairs of students in the test group with values below the minimum of the comparison group. Thus, our approach provides a basis for deciding on which clusters are to be examined further. A limitation of this approach is that we do not know the ground truth in our groups and only be able to back up our reasoning on a comparison.

In summary, we have been successful in providing an opportunity to detect colluding students after the exam. We cannot say if this is sufficient evidence to initialise legal consequences. Nevertheless, we are confident that the higher chance of getting caught has a deterring effect on students. This would be an interesting direction for further research. Moreover, one could collect complementary evidence. By doing so, we found at least two of our suspicious students confirmed.

6. ACKNOWLEDGMENTS

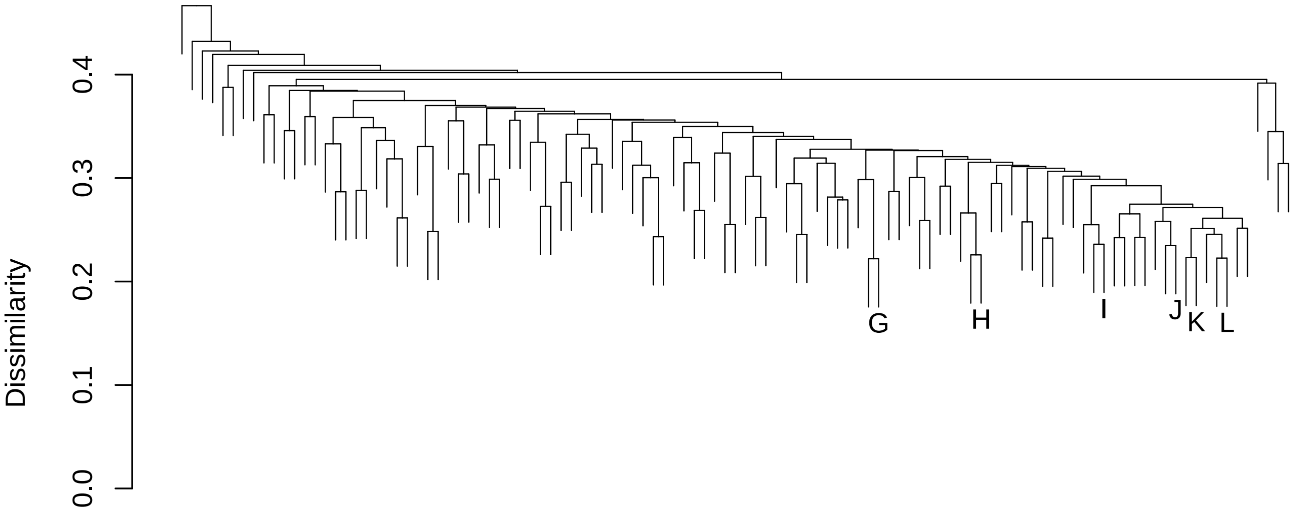


Figure 3: Dendrogram produced by average linkage clustering for the comparison group 2018/19. The dissimilarity of each group's node is displayed on the y-axis. Its value corresponds to the dissimilarity of the group's left and right member. G - L denotes the six lowest dissimilarity clusters.

Part of the work on this project was funded by the German Federal Ministry of Education and Research under grant numbers 01PL16075 and 01JA1910 and by the Foundation for Innovation in University Teaching under grant number FBM2020-EA-1190-00081.

7. ADDITIONAL AUTHORS

Additional authors: Natalie Reckmann (University of Duisburg-Essen, email: `natalie.reckmann@vwl.uni-due.de`).

8. REFERENCES

[1] W. Bowers. Student dishonesty and its control in college, 1964.

[2] R. Chatterjee. Similarity/clustering methods for temporal event data, 2015 (accessed 2021).

[3] C. Cleophas, C. Hoennige, F. Meisel, and P. Meyer. Who's cheating? mining patterns of collusion from text and events in online exams. *Mining Patterns of Collusion from Text and Events in Online Exams (April 12, 2021)*, 2021.

[4] G. Cluskey Jr, C. R. Ehlen, and M. H. Raiborn. Thwarting online exam cheating without proctor supervision. *Journal of Academic and Business Ethics*, 4(1):1–7, 2011.

[5] A. Fask, F. Englander, and Z. Wang. On the integrity of online testing for introductory statistics courses: A latent variable approach. *Practical Assessment, Research, and Evaluation*, 20(1):10, 2015.

[6] D. Goldberg. Programming in a pandemic: Attaining academic integrity in online coding courses. *Communications of the Association for Information Systems*, 48(1):6, 2021.

[7] O. R. Harmon and J. Lambrinos. Are online exams an invitation to cheat? *The Journal of Economic Education*, 39(2):116–125, 2008.

[8] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction.* Springer, 2 edition, 2009.

[9] A. Hellas, J. Leinonen, and P. Ihantola. Plagiarism in take-home exams: Help-seeking, collaboration, and systematic cheating. page 238–243, 2017.

[10] A. Hemming. Online tests and exams: lower standards or improved learning? *The Law Teacher*, 44(3):283–308, 2010.

[11] K. K. Hollister and M. L. Berenson. Proctored versus unproctored online exams: Studying the impact of exam environment on student performance. *Decision Sciences Journal of Innovative Education*, 7(1):271–294, 2009.

[12] M. J. Hussein, J. Yusuf, A. S. Deb, L. Fong, and S. Naidu. An evaluation of online proctoring tools. *Open Praxis*, 12(4):509–525, 2020.

[13] P. Ihantola, A. Vihavainen, A. Ahadi, M. Butler, J. Börstler, S. H. Edwards, E. Isohanni, A. Korhonen, A. Petersen, K. Rivers, et al. Educational data mining and learning analytics in programming: Literature review and case studies. *Proceedings of the 2015 ITiCSE on Working Group Reports*, pages 41–63, 2015.

[14] J. Leinonen, K. Longi, A. Klami, A. Ahadi, and A. Vihavainen. Typing patterns and authentication in practical programming exams. pages 160–165, 2016.

[15] S. Manoharan and X. Ye. On upholding academic integrity in online examinations. In *2020 IEEE*

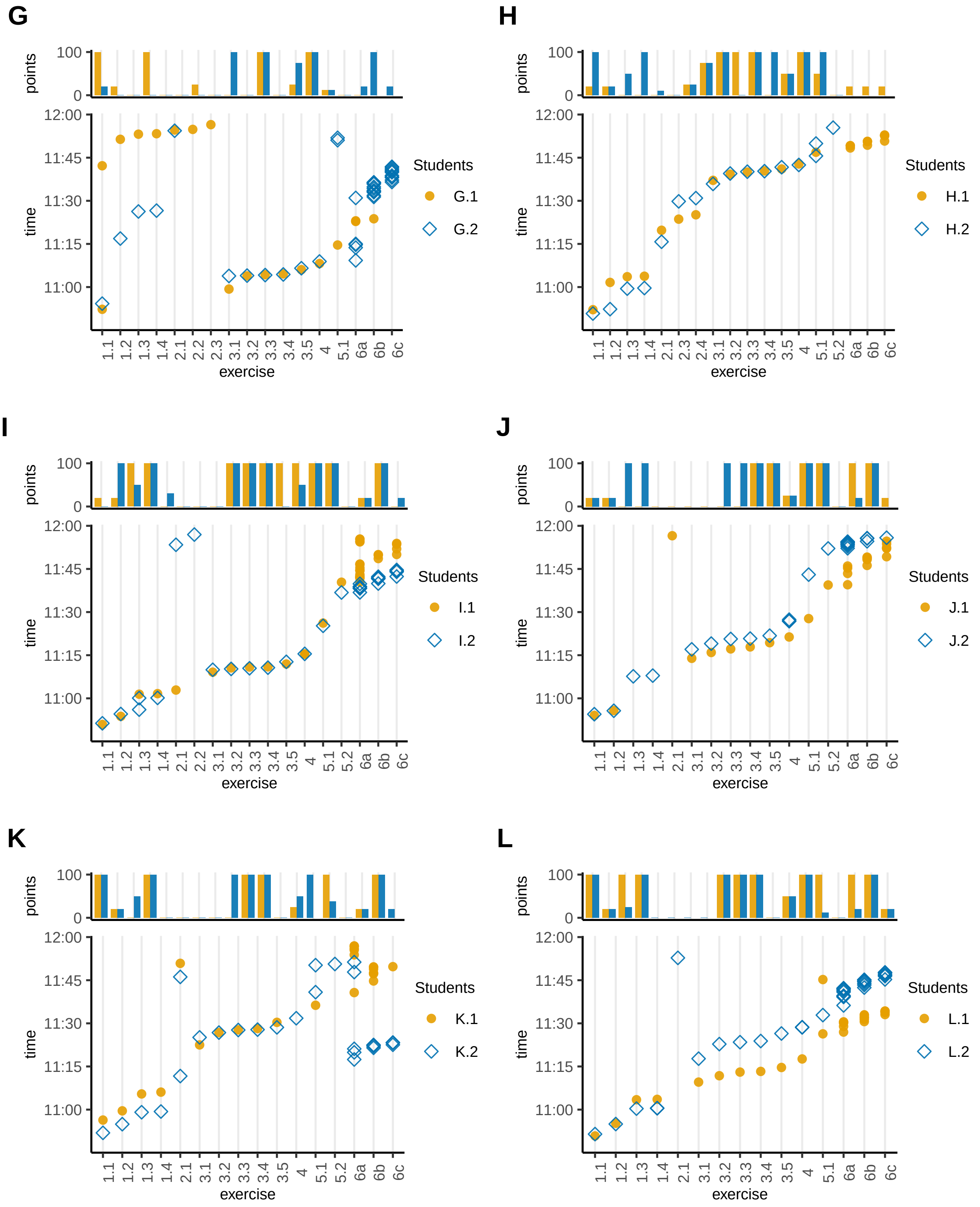
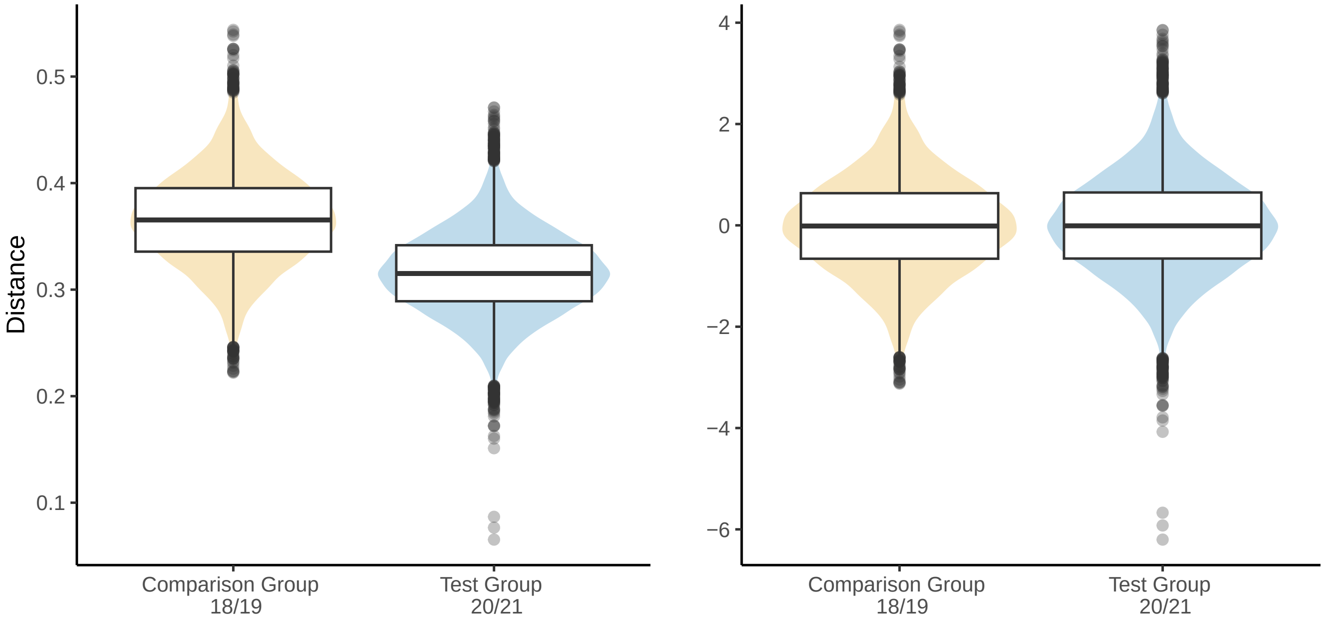


Figure 4: Comparison of the event logs and achieved points for each of the comparison group's (2018/19) six lowest dissimilarity clusters (G - L). The letters of the sub-figures correspond to the marked clusters in Figure 3. The number behind the letters refers to the node's left and right arm, respectively. At the bottom of each sub-figure, the sub-tasks are plotted against the clock time. Above the scatterplot, a bar chart is added to compare the points per subtask.



(a) Non-Normalised

(b) Normalised

Figure 5: Boxplot of the pairwise distance measures.

Conference on e-Learning, e-Management and e-Services (IC3e), pages 33–37. IEEE, 2020.

[16] D. L. McCabe, L. K. Treviño, and K. D. Butterfield. Cheating in academic institutions: A decade of research. *Ethics &Behavior*, 11(3):219–232, 2001.

[17] J. Miltenburg. Online teaching in a large, required, undergraduate management science course. *INFORMS Transactions on Education*, 19(2):89–104, 2019.

[18] M. R. Olt. Ethics and distance education: Strategies for minimizing academic dishonesty in online assessment. *Online journal of distance learning administration*, 5(3):1–7, 2002.

[19] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021.

[20] N. Schwinning, M. Striewe, T. Massing, C. Hanck, and M. Goedicke. Towards digitalisation of summative and formative assessments in academic teaching of statistics. In *Proceedings of the Fifth International Conference on Learning and Teaching in Computing and Engineering*, 2017.

[21] P. C. Shon. How college students cheat on in-class examinations: Creativity, strain, and techniques of innovation, 2006.

APPENDIX

A. VARIABLE DESCRIPTION

Table 2: Variable Description.

Variable	Description
s_{ij}	Points achieved in the jth subtask by the ith student.
v_{ij}	Even patterns for the jth subtask by the ith student.
x_{ij}	Measurement for the ith object and jth attribute. Here, x_{ij} only functions as a variable for explaining the general clustering approach.

B. ATTRIBUTE WEIGHTS

Below we list the attribute weights used for building the global object dissimilarity.

Table 3: The weights for all attributes in the test group (2020/21), rounded to three decimal places.

	weights	
(sub-)tasks	event patterns	points
1.1 - 5.2	0.052	0.013
6a - 6c	0.026	0.013
7	0.026	0.013

Table 4: The weights for all attributes in the comparison group (2018/19), rounded to three decimal places.

	weights	
(sub-)tasks	event patterns	points
1.1 - 5.2	0.045	0.011
6a - 6c	0.022	0.011

C. DENDROGRAM AND COPHENETIC CORRELATION COEFFICIENT

Table 5: The cophenetic correlation coefficient for all three linkage methods for the comparison (2018/19) and test (2020/21) group.

	C	
Linkage method	2020/21	2019/18
single	0.6610	0.6659
complete	0.4424	0.5051
average	0.6964	0.7595

We must consider that clustering algorithms enforce a hierarchical structure on the data. This structure, however, does not have to exist. There exist a good amount of methods to assess how faithfully each algorithm represents the original distances in the data. Here, we use the cophenetic correlation coefficent (C). This is defined as the linear correlation between the pairwise dissimilarity $D(x_i, x_{i'})$ from the original distance matrix and the corresponding *cophenetic* dissimilarity from the dendrogram $t(x_i, x_{i'})$, i.e., the height of the node of the cluster. Let $\overline{D}$ be the mean of $D(x_i, x_{i'})$ and $\bar{t}$ be the mean of $t(x_i, x_{i'})$. Then, C can be written as

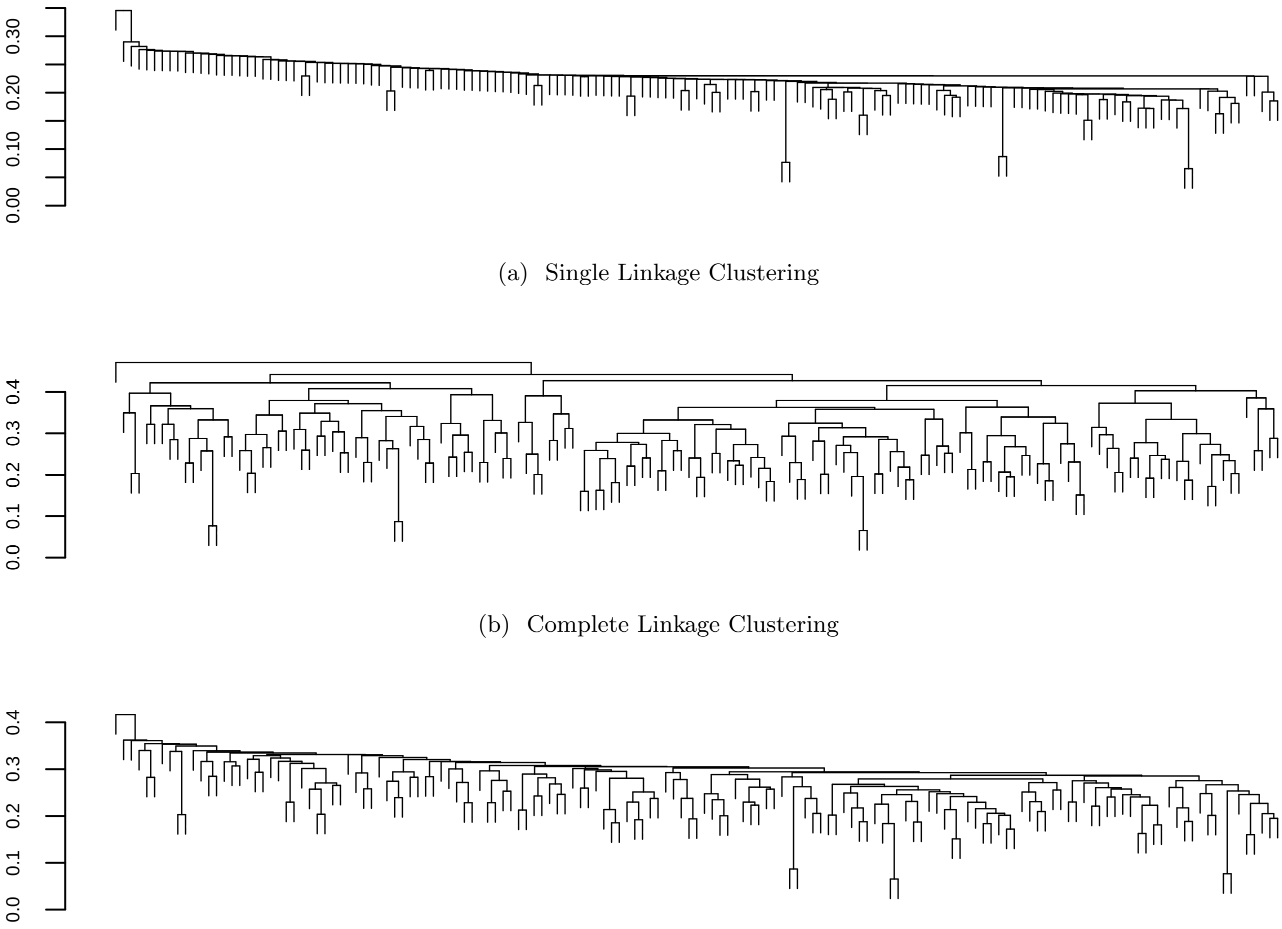
$$C = \frac{\sum_{i<i'} \left(\left(D(x_i, x_{i'}) - \overline{D} \right) \left(t(x_i, x_{i'}) - \bar{t} \right) \right)}{\sqrt{\sum_{i<i'} \left(D(x_i, x_{i'}) - \overline{D} \right)^2 \sum_{i<i'} \left(D(x_i, x_{i'}) - \bar{t} \right)^2}}. \tag{B.1}$$

Table 5 shows C for all three linkage methods. The clustering with the complete linkage method seems to be the most unsuitable. The results of the single and average linkage clustering seem to be an adequate representation, with the latter a slightly better fit. We therefore proceed with the average linkage method in all further steps.

D. ROBUSTNESS CHECK

We repeat our analysis on the same data, but we assign the same weight to each attribute while calculating the global object dissimilarity matrix. In simple terms, we replace the weighted arithmetic mean in equation 3.1 in chapter 3.2 with the ordinary arithmetic mean.

Figure 7 shows the dendrogram for the test data with equal weights. The algorithm still manages to identify the three suspicious clusters. Furthermore, these clusters are still merged at a comparatively low height.



(a) Single Linkage Clustering

(b) Complete Linkage Clustering

(c) Average Linkage Clustering

Figure 6: Comparison of the dendrograms for all three linkage methods in the test group 2020/21.

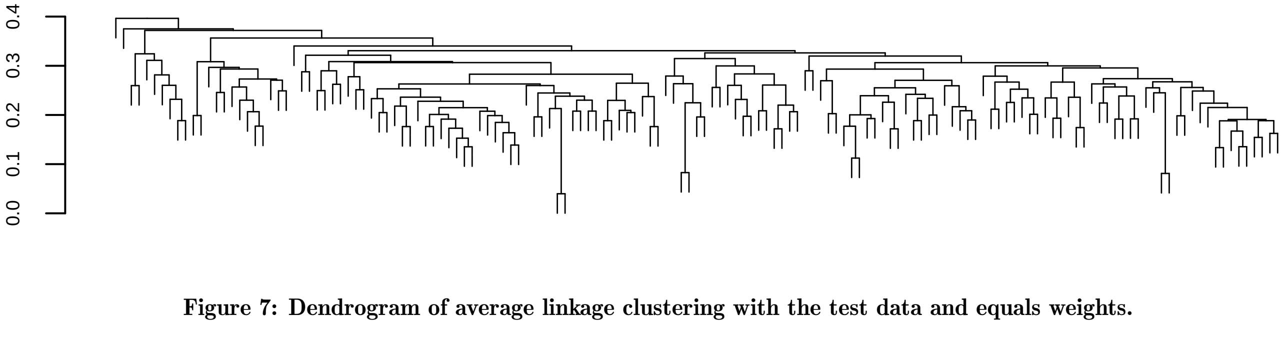


Figure 7: Dendrogram of average linkage clustering with the test data and equals weights.

Automated Search for Logistic Knowledge Tracing Models

Philip I. Pavlik Jr. and Luke G. Eglington
University of Memphis, Amplify Education Inc.
ppavlik@memphis.edu, leglington@amplify.com

ABSTRACT

This paper presents a tool for creating student models in logistic regression. Creating student models has typically been done by expert selection of the appropriate terms, beginning with models as simple as IRT or AFM but more recently with highly complex models like BestLR. While alternative methods exist to select the appropriate predictors for the regression-based models (e.g., stepwise selection or LASSO), we are unaware of their application to student modeling. Such automatic methods of model creation offer the possibility of better student models with either reduced complexity or better fit, in addition to relieving experts from the burden of searching for better models by hand with possible human error. Our new functions are now part of the preexisting R package LKT. We explain our search methods with two datasets demonstrating the advantages of using the tool with stepwise regression and regularization (LASSO) methods to aid in feature selection. For the stepwise method using BIC, the models are simpler (due to the BIC penalty for parameters) than alternatives like BestLR with little lack of fit. For the LASSO method, the models can be made simpler due to the fitting procedure involving a regularization parameter that penalizes large absolute coefficient values. However, LASSO also offers the possibility of highly complex models with exceptional fit.

Keywords

Logistic regression, student modeling, knowledge tracing.

1. INTRODUCTION

Adaptive learning technology requires some way to track a student's learning in order to make decisions about how to interact with the student. The general assumption is that a model of students provides values (e.g., probability estimates typically) that are used to make decisions on pedagogy, the most common decisions being about when or whether to give practice and also how much practice to give (e.g., has the student mastered the proficiency) [13].

This paper describes a tool to build logistic regression models automatically from student data. We focus on finding models that are explainable and parsimonious for a variety of reasons. One reason is because of the needs of open learner models to provide interpretation of the student data, e.g. in a student dashboard, means that there are benefits if it is scrutable, can be made cooperative, and is editable [4]. Complex models make these things more difficult to achieve. Trust is another advantage of explainable systems [10], which can approve adoption by stakeholders.

A common practice in research into student modeling is concerned with choosing models based on fit statistics such as AUC and RMSE. However, the practical benefits of going from an AUC of .85 to .88 (for instance) may be close to zero depending on how the model is being used. If it is being used for reporting proficiency to a dashboard (e.g., in binary terms such as mastered or not), both models may come to the same conclusions. In adaptive instructional systems, whether the better fitting model changes practice sequences depends on the decision rules utilizing the model predictions. Frequently, the same recommended practice sequences will be recommended from both models. In short, there are dramatically diminishing returns from improving model fit, and if the improvement reduces of interpretability and costs 100x more features it likely unjustified. In the present work, we sought to address this tension between optimal model fits and practical considerations.

Unfortunately, because student models differ by content area and the type of learning technology, it often seems necessary to handcraft new models to maximize model accuracy [1, 3, 7, 8, 9, 14, 18, 22]. This has created a parade of alternatives such that a huge amount of researcher knowledge is necessary before a practitioner can easily transfer these methods to new systems. The researcher must be an expert in quantitative methods of knowledge tracing, have a deep understanding of the domain, and understand which learning science principles are important in that domain (repetition, spacing, forgetting, etc.). In addition to these base technical skills there are all the complexities of model building itself such as overfitting and the need for generalization. This base knowledge necessary for model creation creates a long learning curve.

We suppose that the long learning curve in our area can be solved by building better tools to build models. We have been using LKT, which subsumes a large number of prior logistic models by providing a flexible model-building framework in R [15]. However, although LKT enables the use of many predictive features, it doesn't select features for the user. The present work is a demonstration of ongoing work to automatically select a subset of features for the user.

With the excellent model fits of recent deep learning models, some readers will see this prior research as a dead end that people need to move away from, but from these authors' perspective, that is unlikely to be the case. Deep learning student modeling e.g., [17], has been around for several years but can be more complicated to implement within adaptive practice systems than regression and harder to interpret model parameters and interpret errors. New deep learning models can fit well, but do not seem to fit reliably better than simpler alternatives [8]. In many cases, the complexity may be

P. I. P. Jr. and L. G. Eglington. Automated search for logistic knowledge tracing models. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 17–27, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115673

unwarranted for applications unless there is some demonstration that these models can predict student knowledge better than simpler methods like logistic regression.

On the other hand, the simplicity of regression means that software developers and educational content developers can incorporate student models of astounding complexity using basic algebra. Such capability means that incorporating such models as pedagogical decision-makers in educational software is relatively straightforward and has been well described. So, in this paper, we look more deeply at one of the remaining stumbling blocks in the more widespread use of logistic regression to trace student learning.

While the LKT R package allows the application of more than 30 features, it did not previously provide any direction of how to choose these features for a the components (e.g. KCs, students, items) of the data. Choosing such components is also difficult for an expert, since despite an expert perhaps understanding the palette of possible features, given 3 levels of components (as in BestLR) there can be more than 90 possible choices to add to a model (assuming we search across all 30 for each component). Best LR is formulated with the following equation. Where alpha is the student ability, delta is item difficulty, theta is the function log(x+1), beta is the KC difficulty, and gamma and rho capture the effect of prior success and failures for the KC. Sigma transforms the linear measure to the logistic prediction probability.

$$\begin{aligned} BestLR\left(a_{s,t+1} = 1 \middle| q_{s,t+1}, x_{s,1:t}\right) &= \sigma(a_s - \delta_{q_{s,t+1}} + \phi(c_s) + \phi(f_s) \\ &+ \sum_{k \epsilon KC(q_{s,t+1})} \beta_k + \gamma_k \phi(c_{s,k}) + \rho_k \phi\left(f_{s,k}\right)\Big) \end{aligned}$$

Table 1. Start and end stats for each approach with each dataset (BIC and AUC, and par count)

Start Model	Start BIC	End BIC	Start AUC	End AUC	AUC Δ	Parameters Δ
AFM cloze	61563.15	52312.33	0.842	0.858	0.016	-607
BestLR cloze	61001.05	52227.29	0.856	0.862	0.006	-715
Empty cloze	75977.87	52544.5	0.5	0.861	0.361	139
AFM MATHia	51480.63	45602.94	0.811	0.816	0.005	-502
BestLR MATHia	50728	45209.36	0.831	0.822	-0.009	-610
Empty MATHia	59296.01	45611.19	0.5	0.816	0.316	14

To address this problem in the inefficiency of logistic regression modeling, we here describe and test our tool for stepwise student model search in LKT. For the expert, this will save either the need to use cookie cutter models that they know, but that may not be appropriate or the countless hours of manual search that is often necessary when trying to understand modeling in a new domain. For the practitioner the this LKT package update will allow fast creation of models tailored to multiple purposes and domains, saving time and likely opening the possible options. For the student student modeler, the updated package provides a way to begin building models quickly and with sufficient feedback so as to think deeply about the functioning of those models. The example vignette in the LKT package shows many examples from this paper.

2. METHODS

2.1 Stepwise

In the function, the user may set the objective function (BIC, AIC, AUC, R2, or RMSE), but these behave quite similarly in our testing except for BIC, which corrects heavily for the potential of overfitting due to high parameter counts. The user may specify forward or backward search or alternate between forward and backward (bidirectional search). The user also has control over the initial features and components in the model, allowing the exploration of theoretical hypotheses for completed models and the optimization of those models. For example, in our tests, we illustrate starting with the BestLR model and then allowing the algorithm to simplify the model while simultaneously add a key new predictor. The user can also specify the forward and backward step size needed in terms of the objective function (fit statistic) which is also chosen.

2.2 LASSO

An alternative approach to stepwise regression is LASSO regression, a form of regularization. In this method, a penalty term is added to the loss function equal to the sum of the absolute values of the coefficients times a scalar lambda. This penalty term may result in the best fitting model having fewer features if they are correlated. Larger lambda values will result in fewer features. A common method to use this approach is to attempt a large number of potential lambda values, and choose the value with the best cross-validated performance. In the present case, we are particularly concerned with finding interpretable models that are easier to implement, and so larger values that may have slightly worse performance may be preferred. To evaluate the resultant models from LASSO we began by using the glmnet R package to fit both datasets with 100 values starting at the lowest value that would reduce all coefficients to zero (the maximum lambda) decreasing in increments of .001 (the default strategy with glmnet, [6]. At each step, 25-fold cross-validation was performed. This allowed us to evaluate the stability of the candidate lambda values. Subsequent model fitting and analyses used specific lambdas intended to evaluate the fit and interpretability of LASSO models with varying levels of complexity to determine the usefulness of LASSO in comparison to stepwise regression. An important distinction between LASSO and the stepwise approach employed in this work is that for lasso the coefficients for individual KCs may be dropped. For instance, if two different KCs are essentially redundant a LASSO model may reduce a coefficient for one of them to zero if the lambda value is large enough. In contrast, the stepwise regression approach we employed treats the KC model as a single feature, it is either included or it is not.

For nonlinear features logitdec, propdec, and recency, features were generated with parameters from .1 to 1 in .1 increments (e.g., propdec with decay parameters .1, .2, up to 1). All the resultant features were included in the LASSO models to allow us to evaluate which parameter values remained and whether more than one was beneficial.

3. RESULTS

3.1 Bidirectional Stepwise Method

For the stepwise method, it is possible to use any collection of features as a “start” model that is subsequently added to and subtracted from. Using different starts helps us understand how the method can have problems with local minima but also helps us see that these problems are rather minimal as the different starts converge on similar results. At the same time, showing how the method improves upon “stock” models is an important part of the demonstration, showing that these “stock” models are not found to be particularly precise, and we might question whether better local minima are actually an improvement.

Table 2. AFM start results, cloze data.

R^2	params	BIC	AUC	RMSE	action
0.287	676	61563.15	0.842	0.401	starting model
0.354	678	56461.99	0.872	0.380	add: recency-KC..Default.
0.352	643	56251.40	0.871	0.381	drop: intercept-KC..Cluster.
0.362	644	55532.05	0.876	0.377	add: logsuc-CF..Correct.Answer.
0.356	580	55275.79	0.873	0.379	drop: lineafm$-CF..Correct.Answer.
0.351	544	55247.92	0.871	0.381	drop: lineafm$-KC..Cluster.
0.358	546	54698.11	0.874	0.379	add: recency-KC..Cluster.
0.284	69	55130.90	0.840	0.403	drop: intercept-Anon.Student.Id
0.326	71	51965.35	0.860	0.389	add: propdec-Anon.Student.Id
0.321	69	52312.33	0.858	0.391	drop: recency-KC..Cluster.

Table 3. BestLR start results, cloze data.

R^2	params	BIC	AUC	RMSE	action
0.319	849	61001.05	0.856	0.392	starting model
0.371	851	57098.22	0.879	0.374	add: recency-KC..Default.
0.337	374	54422.55	0.865	0.386	drop: intercept-Anon.Student.Id
0.337	303	53650.51	0.865	0.386	drop: intercept-KC..Default.
0.336	267	53327.24	0.865	0.386	drop: logfail$-KC..Cluster.
0.331	203	53063.81	0.862	0.388	drop: logfail$-CF..Correct.Answer.
0.328	168	52849.33	0.861	0.389	drop: intercept-KC..Cluster.
0.325	132	52731.07	0.859	0.390	drop: logsuc$-KC..Cluster.
0.332	134	52227.29	0.862	0.388	add: recency-KC..Cluster.

Table 4. Empty start results, cloze data.

R^2	params	BIC	AUC	RMSE	action
0.000	1	75977.87	0.500	0.498	null model
0.174	65	63428.69	0.746	0.440	add: logsuc$-CF..Correct.Answer.
0.219	67	60095.24	0.788	0.425	add: recency-KC..Default.
0.282	138	56024.84	0.839	0.404	add: intercept-KC..Default.
0.328	140	52544.50	0.861	0.389	add: propdec-Anon.Student.Id

We choose to use AFM [1] and BestLR [8] models as starting points, in addition to using an empty start (which included a global intercept to account for the grand mean of performance, as did all our models without explicit intercepts). AFM and BestLR starts are interesting since they illustrate the advantages of using the search method by arriving at models that fit better or equivalently with fewer parameters. Furthermore, using these start points allows us to show that these canonical models are not even local minima, which highlights how our methods are useful. If these models are particularly strong, it should not be possible to add terms to them, and the current terms should not be dropped. See Table 1 for summary.

Using these starts we search over a preset group of features that is meant to be “complete enough” to produce interesting relevant results and goes beyond BestLR features (which it includes), to also include some of the simplest and most predictive non-linear features we have developed in other work [15].

We used several features, which we crossed with all the possible components (listed below) for each dataset. A $ indicates that the feature is fit with 1 coefficient per level of the component (e.g., one coefficient for each KC, student, or item). Intercept (a fixed coefficient for each level of the feature) does not require the $ notation since it is always fit this way. In contrast, without a $ indicates that all levels of the KC behave the same, so for example lineafm$ for the student means that there would be a continuous linear increase in performance for each trial for each student, with a different rate for each student.

We choose a limited set of likely features from the LKT software to search across. These included

- Intercept–one coefficient for each level of the component factor
- Lineafm–one coefficient to characterize the linear change with each repletion of the component
- Logafm– one coefficient to characterize the logarithmic change with each repetition for each level of the component. 1 is added to prior repetitions.
- Logsuc– one coefficient to characterize the logarithmic change with each successful repetition for each level of the component. 1 is added to prior repetitions.
- Logfail– one coefficient to characterize the logarithmic change with each failed repetition for each level of the component. 1 is added to prior repetitions.
- Linesuc– one coefficient to characterize the linear change with each successful repetition for each level of the component

Table 5. AFM start results, MATHia data.

R^2	params	BIC	AUC	RMSE	action
0.226	517	51480.64	0.811	0.390	starting model
0.247	519	50275.13	0.823	0.384	add: recency-KC..MATHia.
0.163	20	49815.16	0.771	0.408	drop: intercept-Anon.Student.Id
0.227	22	46094.29	0.812	0.390	add: logitdec-Anon.Student.Id
0.224	13	46120.77	0.810	0.391	drop: lineafm$-KC..MATHia.
0.234	15	45602.94	0.816	0.388	add: logitdec-KC..MATHia.

Table 6. BestLR start results, MATHia data.

R^2	params	BIC	AUC	RMSE	action
0.258	626	50728.00	0.831	0.381	starting model
0.275	627	49762.34	0.840	0.375	add: linesuc-Problem.Name
0.241	128	46362.18	0.822	0.385	drop: intercept-Anon.Student.Id
0.252	130	45749.90	0.828	0.382	add: recency-KC..MATHia.
0.240	32	45377.63	0.821	0.385	drop: intercept-Problem.Name
0.250	34	44841.98	0.827	0.383	add: recency-Problem.Name
0.246	25	44959.37	0.825	0.384	drop: logfail$-KC..MATHia.
0.240	16	45209.36	0.822	0.386	drop: logsuc$-KC..MATHia.

Table 7. Empty start results, MATHia data.

R^2	params	BIC	AUC	RMSE	action
0.000	1	59296.01	0.500	0.452	null model
0.161	3	49773.24	0.768	0.409	add: logitdec-KC..MA-THia.
0.189	11	48207.27	0.787	0.402	add: intercept-KC..MA-THia.
0.218	13	46506.96	0.806	0.393	add: propdec-Anon.Student.Id
0.233	15	45611.19	0.816	0.388	add: recency-KC..MA-THia.

- Linefail– one coefficient to characterize the linear change with each failed repetition for each level of the component
- Logitdec–one coefficient to characterize the logit of prior success and failures for the component (seeded with 1 success and 2 failures resulting in a start value of 0, e.g. log(.5/.5)=0). Uses a nonlinear exponential decay to weight priors according to how far they are back in the sequence for the component traced.
- Propdec–one coefficient to characterize the probability of prior success and failures for the component (seeded with 1 success and 2 failures resulting in a start value of 0, e.g. .5/1)=.5). Uses a nonlinear exponential decay to weight priors success and failures according to how far they are back in the sequence for the component traced.
- Recency– one coefficient to characterize the influence of the recency of the previous repetition only, where t is the time since the prior repetition at the time of the new prediction and d characterize non-linear decay. The value is computed as t^{-d}.
- Logsuc$–like logsuc above, except one coefficient is added per level of the component (e.g., different effects for each KC or item)
- Logfail$– like logfail above, except one coefficient is added per level of the component (e.g., different effects for each KC or item)

3.1.1 Cloze practice

The statistics cloze dataset included 58,316 observations from 478 participants who learned statistical concepts by reading sentences and filling in missing words. Participants were adults recruited from Amazon Mechanical Turk. There were 144 KCs in the dataset, derived from 36 sentences, each with 1 of 4 different possible words missing (cloze items). The number of times specific cloze items were presented was manipulated, as well as the temporal spacing between presentations (narrow, medium, or wide). The post-practice test (filling in missing words) could be after 2 minutes, 1 day, or 3 days (manipulated between students).

The stimuli type, manipulation of spacing, repetition of KCs and items, and multiple-day delays made this dataset appropriate for evaluating model fit to well-known patterns in human learning data (e.g., substantial forgetting across delays, benefits of spacing). The dataset was downloaded from the Memphis Datashop repository.

As components we choose to use the ids for the student (Anon.Student.Id), sentence itself (KC..Cluster, 32 levels due to each sentence having 2 feedback conditions which we do not investigate here), specific items (KC.Default.) and the response word (CF..Correct.Answer.). KC..Default. and CF..Correct.Answer. had a good deal of overlap with KC..Default. since there were 72 items with 64 different responses. Here are two examples of these items, "The standard deviation is a __________ that describes typical variability for a set of observations.", and "Standard deviation is the __________ of the variance, also known as root mean squared error."

Tables 2, 3 and 4 show the results for the different start models.

For the AFM start the final model is specified in feature(component) notation, see equation below. See Table 2 and Figure 1 for the step actions that led to this final model.

$$intercept(CF..Correct.Answer.) + recency(KC..Default.) \\ + logsuc(CF..Correct.Answer.) \\ + propdec(Anon.Student.Id)$$

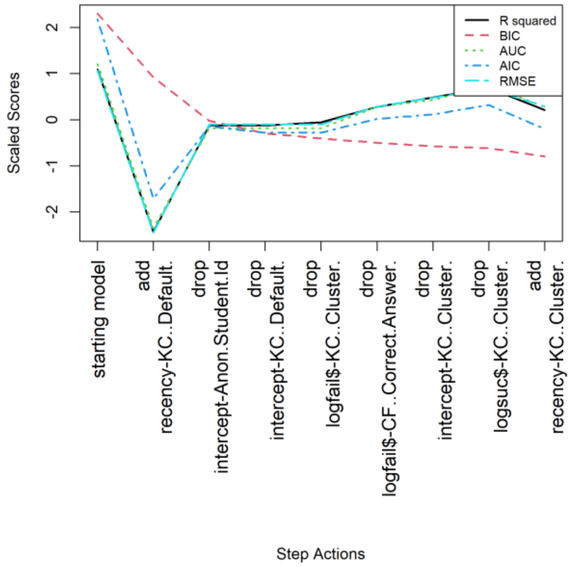


Figure 1. Scaled fit statistic (Z-score) changes during BIC bidirectional stepwise search for AFM model start with Cloze data.

For the BestLR start the final model is specified in feature(component) notation, see equation below. See Table 3 and Figure 2 for the step actions that led to this final model.

$$logsuc(Anon.Student.Id.) + logfail(Anon.Student.Id.) \\ + intercept\ (CF..Correct.Answer.) \\ + logsuc\$(CF..Correct.Answer.) \\ + recency(KC.Default) \\ + recency(CF..Correct.Answer.)$$

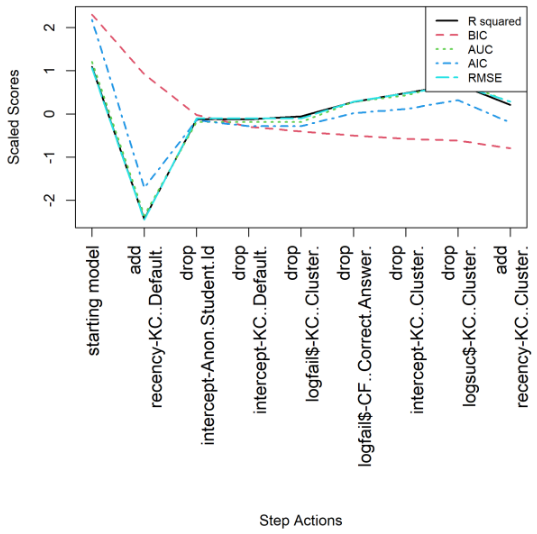


Figure 2. Scaled fit statistic (Z-score) changes during BIC bidirectional stepwise search for BestLR model start with Cloze data.

For the empty start the final model is specified in feature(component) notation, see equation below. See Table 4 and Figure 3 for the step actions that led to this final model.

$$\begin{aligned} &\mathit{logsuc}\$(\mathit{CF..Correct.Answer.}) + \mathit{recency}(\mathit{KC..Default.}) \\ &\quad + \mathit{intercept}(\mathit{KC..Default.}) \\ &\quad + \mathit{propdec}(\mathit{Anon.Student.Id}) \end{aligned}$$

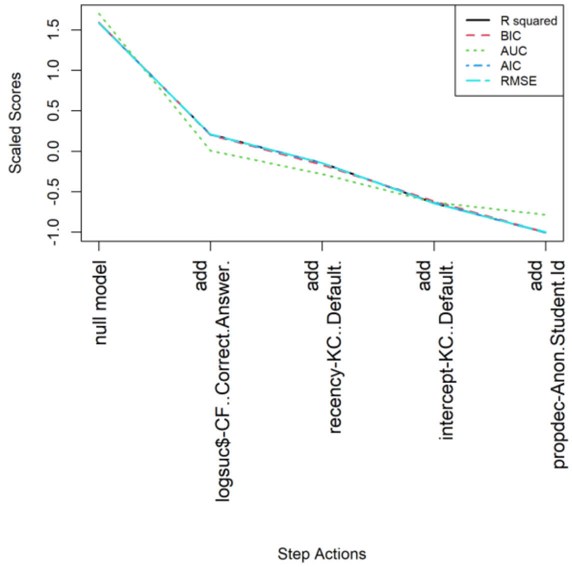


Figure 3. Scaled fit statistic (Z-score) changes during BIC bidirectional stepwise search for empty model start with Cloze data.

3.1.2 *MATHia Cognitive Tutor equation solving*

The MATHia dataset included 119,379 transactions from 500 students from the unit Modeling Two-Step Expressions for the 2019-2020 school year. We used the student (Anon.Student.Id), MATHia assigned skills (KC..MATHia.), and Problem.Name as the item. This meant that our item parameter was distributed across the steps in the problems. There were 9 KCs and 99 problems. We chose not to use the unique steps as an item in our models for simplicity. This dataset included skills such as such as "write expression negative slope" and "enter given, reading numerals".

Tables 5, 6 and 7 show the results for the different start models.

For the AFM start the final model is specified in feature(component) notation, see equation below. See Table 5 and Figure 4 for the step actions that led to this final model.

$$\begin{aligned} &\mathit{logitdec}(\mathit{Anon.Student.Id}) + \mathit{logitdec}(\mathit{KC..MATHia.}) \\ &\quad + \mathit{recency}(\mathit{KC..MATHia.}) \\ &\quad + \mathit{intercept}(\mathit{KC..MATHia.}) \end{aligned}$$

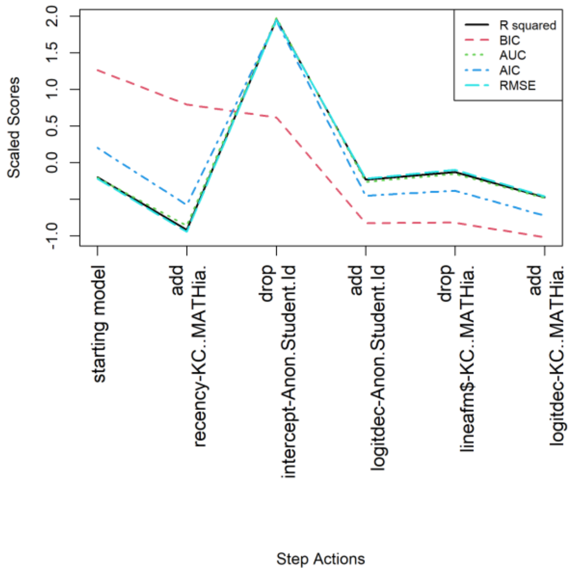


Figure 4. Scaled fit statistic (Z-score) changes during BIC bidirectional stepwise search for AFM model start with MATHia data.

For the BestLR start the final model is specified in feature(component) notation, see equation below. See Table 6 and Figure 5 for the step actions that led to this final model.

$$\begin{aligned} &\mathit{logfail}(\mathit{Anon.Student.Id}) + \mathit{logsuc}(\mathit{Anon.Student.Id}) \\ &\quad + \mathit{intercept}(\mathit{KC..MATHia.}) \\ &\quad + \mathit{recency}(\mathit{KC..MATHia.}) \\ &\quad + \mathit{recency}(\mathit{Problem.Name}) \\ &\quad + \mathit{linesuc}(\mathit{Problem.Name}) \end{aligned}$$

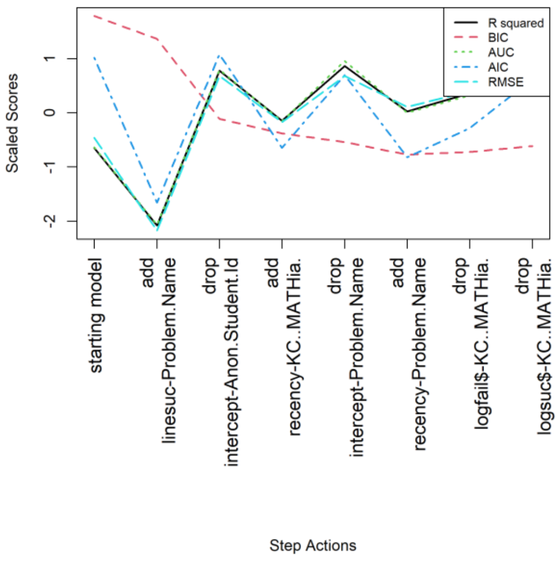


Figure 5. Scaled fit statistic (Z-score) changes during BIC bidirectional stepwise search for BestLR model start with MATHia data.

For the empty start the final model is specified in feature(component) notation, see equation below. See Table 7 and Figure 6 for the step actions that led to this final model.

$$propdec(Anon.Student.Id) + intercept(KC..MATHia.) \\ + recency(KC..MATHia.) \\ + logitdec(KC..MATHia.)$$

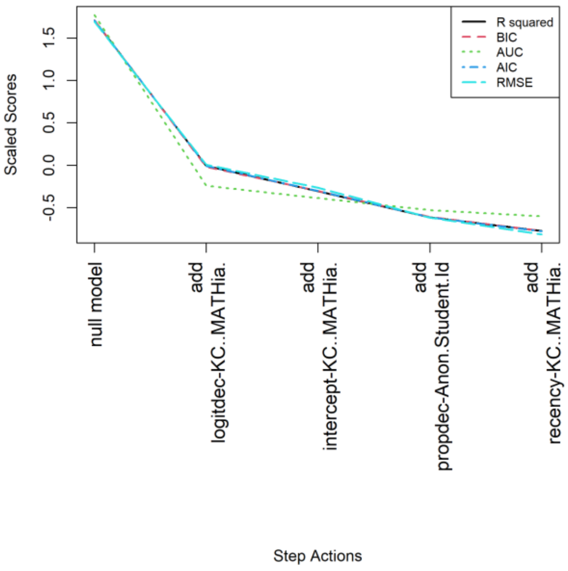


Figure 6. Scaled fit statistic (Z-score) changes during BIC bidirectional stepwise search for empty model start with MATHia data.

3.2 Cloze with stepwise

For the cloze dataset, the models from the 3 starting points produce somewhat different results, illustrating the problem with any stepwise method due to it not being a global optimization. However, considering the fact that our goal is to practically implement these models, the result also suggests a solution to this local minima problem. By using more than one starting point we can identify the essential feature that explain the data.

For example, in these cloze results note that the recency feature used for the KC-Default is particularly predictive. In this dataset simply means that the time since the last verbatim repetition (KC Default) was a strong predictor with more recent time since last repetition leading to higher performance.

Successes were also important, but curiously they matter most for the KC-Correct-Answer. In all case the log of the success is the function best describing the effect of the correct responses. In this dataset, this mens that each time they responded with a fill-in word and it was correct, they would be predicted to do better the next time that word was the response. The log function is just a way to bias the effect of successes to be stronger for early succeses.

While the recency being assigned to the exact repetitions (KC-Default) indicates the importance of memory to performance, the tracking of success (as permanent effects) across like responses suggests that people are actually learning the vocabulary despite showing forgetting.

Consistent across all three final models is also the attention to student variability modeling. In BestLR, the log success and failure predictors for the student in the model mean that the student intercept is removed in an elimination step as redundant (this is also due to the BIC method, which penalizes the student intercept as unjustifiably complex). Interestingly, in the AFM and empty start models, we find that the propdec feature is added to capture the student variability after the intercept is removed, since there starts did not trace student performance with their start log success and failure feature as did BestLR from the start. The MATHia data has the same "problem" with BestLR start due to BestLR serving as enough of a local minima to block the addition of terms. More on this in the limitations section. Practically these features are important, since they allow the model to get an overall estimate for the student that greatly improves prediction of individual trials.

In summary, there appears to be no great advantage to starting with a complex starting model. Indeed, in all cases the stepwise procedure using BIC greatly simplifies the models by reducing the number of coefficients. It appears that prior models produced by humans (in this case, AFM and BestLR) do not produce better results in the model space than simply starting with an empty null hypothesis for the model. Furthermore, all three start models result in final models have no fixed student parameters, so should work for new similar populations without modifications, unlike AFM and BestLR which relied on fixed student intercepts

3.3 MATHia with stepwise

Practically speaking for the MATHia case we also see the importance of student variability, recency, and the correctness at fine grain by KC and item for all the models. Digging into the detail, ewe can see the BestLR start has some effect on the quantitative fit and chosen model. Most notably, while AFM and empty starts result in the student intercept being dropped in favor of logitdec and propdec respectively, the BestLR start retains the log success and failures predictors for the student. At the same time, Best LR, perhaps because it begins with the Problem.Name intercept as a covariate, adds features more features for Problem.Name, such as linesuc and recency. It seems clear that BestLR causes a different result. At this time, we might favor the simpler results of AFM or empty starts, but consider that the BestLR start fits the data by AUC slightly better than the BestLR result. This implies that AFM and empty starts are simply producing overly simplistic results. Consider we only dropped and removed terms when BIC gain was at least 500. We expect that running from an empty start to a lower BIC threshold would result in more commonality with BestLR start. To test this, we ran the empty start and indeed we found that the result became more similar to the BestLR result with the addition of linesuc and recency for the problems. Since this model (shown below) is still slightly worse than the BestLR start it implies that the algorithm favors composite features despite better fit from individual features (logsuc and logfail). We discuss this in the limitations and future work sections.

$$\begin{aligned} propdec(Anon.Student.Id) &+ intercept(KC..MATHia.) \\ &+ recency(KC..MATHia.) \\ &+ logitdec(KC..MATHia.) \\ &+ linesuc\ (Problem.Name) \\ &+ receny(Problem.Name) \end{aligned}$$

Finally, all the models retained an intercept for the KCs, and all of the models capture MATHia KC performance change with the logitdec feature.

3.4 LASSO Method

A primary goal of the LASSO analyses is to determine how well the approach can inform a researcher about which features are most important, and guide the researcher toward a fairly interpretable, less complex, but reasonably accurate model. See Figures 7 and 8 plotting the relationship between the number of features and AUC across 50 values of lambda ranging from .192 (a large penalty) to .0001 (very small penalty). For both datasets, there is clearly diminishing fit benefits as the number of features is increased (from a smaller lambda). Both curves have clear inflection points. At the inflection point, the coefficients for most features have been dropped to zero (see Table 8). Note in Table 8 that the MATHia dataset in particular fits quite well without many parameters for individual KCs. What remains appear to be the more robust and important features.

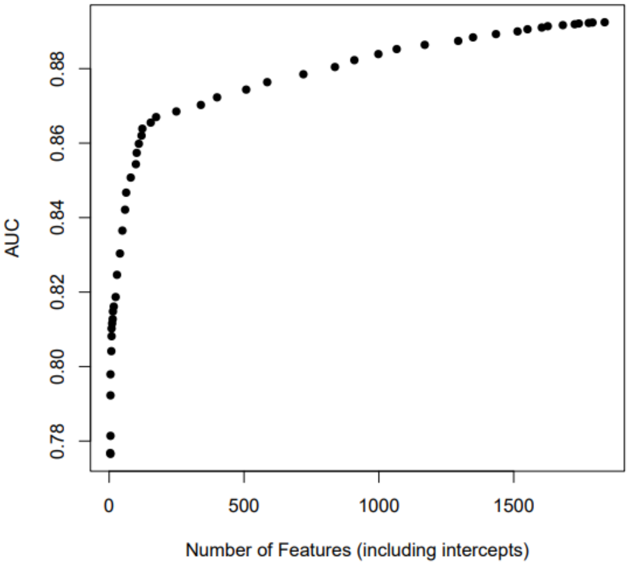


Figure 7. AUC for cloze dataset as a function of the number of retained features. There is a clear elbow at AUC = ~.86 with 123 features (including KC intercepts) beyond which there are diminishing returns. For comparison, BestLR was .856 with 849 coefficients.

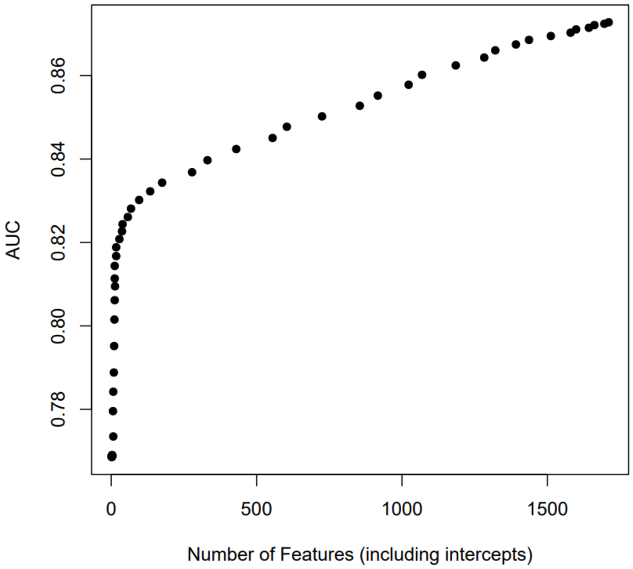


Figure 8. AUC for MATHia dataset as a function of the number of retained features. There is a clear elbow at AUC = ~.82 with 29 features (including KC intercepts) beyond which there are diminishing returns. For comparison, BestLR was .831 with 626 coefficients.

Table 8. Proportion of features with nonzero coefficients in Lasso model at AUC inflection points in Figures 7 and 8.

Feature	Cloze	MATHia
KC intercepts	.4069	.1111
KC logsuc	.0116	.027
KC logfail	.1104	0
Student Intercept	.0083	.002
Student logsuc	0	0
Student logfail	.002	0

The final features that remained for LASSO models near the inflection points partially overlapped with those found with our stepwise regression approach as expected. Below the top 10 features for each dataset are listed in order of relative importance (see Tables 9, 10, and 11 below). When the results didn't agree with the stepwise results, it appears that it may be because a stricter LASSO should be employed. For instance, a recency feature for the Problem.Name KC with decay parameter .1 remained in the MATHia dataset. However, it has a negative coefficient, which is challenging to interpret given that the negative sign implies correctness probability increases as time elapses. A larger lambda value may be justified.

For the Cloze dataset, a large number of features remained even at the inflection point (123) and they were missing many features we stepwise added. Inspecting the 123 features we saw that the variance stepwise captured in single terms was distributed across many terms in LASSO. Given that one goal of this work is to make simpler and more easily interpretable models for humans, we tried a larger penalty to reduce the number of features to 24. The resulting top 10 are in Table 10. This model is more interpretable to a human, with mostly recency features, recency-weighted proportion features, and counts of success for KC. While the match to stepwise is not exact we can now see it attending to student and KC successes and failures with features like logitdec in the top 10. It appears that lambda values are a sort a human interpretability index. Larger values make the resultant models more human interpretable, and in this case still create well-fitting models. Overall, the agreement between the approaches is encouraging evidence that these methods may be useful for researchers.

Table 9. Top 10 features in Cloze model at inflection point near AUC = .86. Bolded features were also in the final empty start stepwise regression model.

Feature	Standardized coefficient	Feature Type
RecencyKC..Default_0.4	4.4072	Knowledge Tracing
KC..Default._1	1.4984	Intercept
KC..Default._2	-1.4870	Intercept
KC..Default._3	-1.3911	Intercect
KC..Default._4	-1.3922	Intercept
KC..Default._5	-1.2857	Intercept
recencyKC..Cluster.0.2	0.9533	Knowledge Tracing
KC..Default._6	0.9834	Intercept
CF..Correct.Answer._1	-0.9438	Intercept
KC..Default._7	0.8427	Intercept

Table 10. Top 10 features in Cloze model when a larger lambda is imposed to reduce the total number of features to 24. Resulting AUC = .818. Bolded features were also in the final empty start stepwise regression model.

Feature	Standardized coefficient	Feature Type
recencyKC..Default._0.3	1.8569	Knowledge Tracing
recencyCF..Correct.Answer._0.2	1.4381	Knowledge Tracing
recencyKC..Cluster._0.3	0.7548	Knowledge Tracing
recencyAnon.Student.Id_0.1	-0.6053	Knowledge Tracing
logsucKC..Default**.**	0.4211	Knowledge Tracing
logitdecCF..Correct.Answer._0.9	0.3766	Knowledge Tracing
logitdecAnon.Student.Id_1	0.2204	Knowledge Tracing
recencyKC..Default._0.2	0.3870	Knowledge Tracing
KC..Default._2	-0.1350	Intercept
logitdecAnon.Student.Id_0.9	0.1343	Knowledge Tracing

Table 11 Top 10 features in MATHia model at inflection point near AUC = .82. Bolded features were also in the final empty start stepwise regression model.

Feature	Standardized coefficient	Feature Type
recencyKC..MATHia._0.2	1.1103	Knowledge Tracing
KC..MATHia._1	1.1777	Intercept
KC..MATHia._2	1.0789	Intercept
KC..MATHia._3	0.9621	Intercept
recencyKC..MATHia._0.3	0.9831	Intercept
recencyProblem.Name_0.1	-0.4848	Knowledge Tracing
KC..MATHia._4	-0.4080	Intercept
logitdecKC..MATHia._0.9	0.3138	Knowledge Tracing
Problem.Name_2	-0.2098	Intercept
logitdecAnon.Student.Id_0.9	0.1682	Knowledge Tracing

If minimum BIC is used instead of the AUC inflection point, the "optimal" models have slightly more features (e.g., 154 for cloze instead of 123, and 96 instead of 29 for MATHia), but still far fewer than the full model. The BIC minimum as a function of features is displayed in Figures 9 and 10.

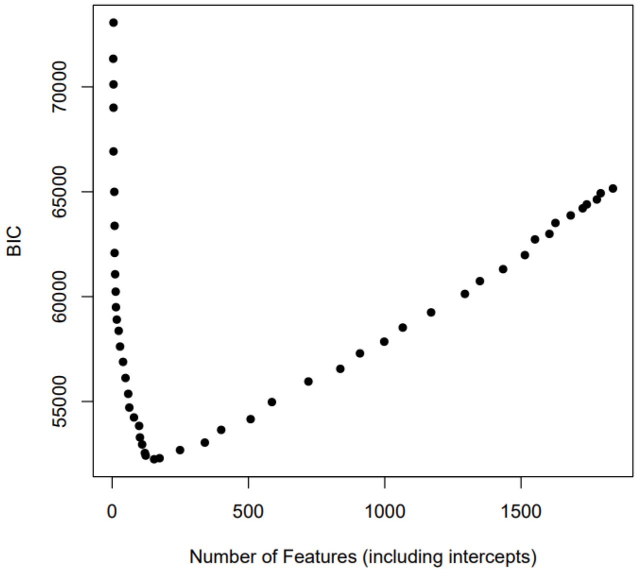


Figure 10. BIC as a function of number of features in Lasso model with cloze dataset. Minimum BIC has 154 features including intercepts, with AUC = .8655 and RMSE = .3868.

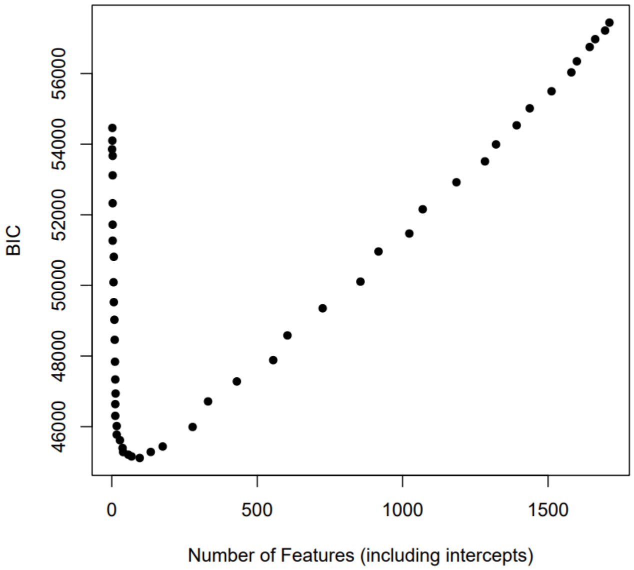


Figure 11. BIC as a function of number of features in Lasso model with MATHia dataset. Minimum BIC has 96 features including intercepts, with AUC = .8301 and RMSE = .3816.

Below in Table 12 we provide a final contrast of three example models with small, medium, and large lambda penalty terms. Interestingly, both datasets can achieve approximately AUC = .80 with ~10 features! Intercepts are considered features in this case, which means that a majority of the KC and student intercepts were dropped. This highlights a potential benefit of the LASSO approach for evaluating the fit of the KC model. It also suggested that student intercepts may not always be necessary in the presence of features like logitdec and propdec, which can stand in for intercepts to adjust for student differences.

Table 12. Lasso model fits with three levels of regularization. The strictest (fewest parameters) demonstrates how well a much smaller model can perform if needed. The medium-level model is the model at the AUC inflection points in Figures 10 and 11. The least strict model demonstrates the diminishing returns of increased parameters.

Dataset	R^2	N params	BIC	AUC	RMSE
MATHia	.2955	10	48459.54	.7951	0.4011
MATHia	.3396	28	45617.33	.8208	0.3871
MATHia	.3805	555	47884.99	.8450	0.3730
Cloze	.2024	11	61065.81	.8115	0.4310
Cloze	.3328	123	52419.37	.8638	0.3882
Cloze	.3587	508	54157.31	.8743	0.3795

4. DISCUSSION

The results suggest both stepwise and LASSO methods work excellently to create, improve, and simplify student models using logistic regression. Both approaches generally agreed that recency features, logsuc, and recency-weighted proportion measures like propdec and logitdec were important. They also agreed that the number of necessary features was substantially less than the full models. While some have argued against stepwise methods [19], we think that stepwise methods worked relatively well here because the feature choices were not arbitrary. We did not simply feed in all the features we could find. Instead, we choose a set of features that can be theoretically justified.

Interpreting the results from these models needs to begin from consideration of the individual features. Each individual feature being found for a model means that the data is fit better if we assume the feature is part of the story for learning in the domain the data comes from. Clearly, we might expect different features for different domains of learning, and practically, knowing the features predicting learning means that we can better understand the learning better. For example, knowing that recency is a factor, or knowing that overall student variability is a big effect. The models this system builds might simply be used to understand online learning in some domain, but the expert building instruction technology might also use them in an adaptive learning environment to make decisions about student pedagogy or instruction.

4.1 Limitations and Future Work

A primary limitation of the present work is that we only included a subset of the features that are already known and established theoretically. There were also known features we did not include (e.g., specific time window features and interactions among features). An extension of this work will be to include more features as well as a step to generate and test novel features that may be counterintuitive. For instance, KC model improvement algorithms could be incorporated into the process [12]. However, how much variance is left to explain that is not covered by the set of features we used? With both datasets, models with AUC > .8 were found using only a relatively small subset of the potential features. Some fraction of unexplained variance is always to be expected due to inherent noise, KC model errors, and measurement error. A significant amount of remaining variance may be individual student differences that justify *different types of models* that update automatically to attempt to estimate individual learning rates, for example. These approaches are beyond the scope of this paper but an important topic for future work.

An opportunity for future work may also be to use these features as components in other model architectures, such as Elo or deep learning approaches. There are ongoing efforts to make deep learning models more interpretable, but for the present work we focused on a model architecture that is relatively interpretable to non-experts, logistic regression. Elo modeling is also particularly promising due to its simplicity and self-updating function [16]. Elo can be adjusted to include KT features like counts of successes and failures [11], but standard Elo without KT features also serves as a strong null model since it does reasonably well without KT features.

A key limitation of the stepwise method is the individuality of features. This is illustrated by the way that logsuc and logfail are retained in the BestLR MATHia model, but they are not added in any of the other models. Their retention in BestLR, may be best, but it may also reflect the standard tendency of stepwise methods to block the addition of new terms (possibly better) that are collinear with prior terms. This may be unavoidable, but also an uncommon problem we think. In contrast the fact that logsuc and logfail are not added for the student when nothing is already present might be because this requires 2 steps of the algorithm, while adding composite features like propdec or logitdec requires 1 step. Since stepwise selection is based on a greedy step optimization it ignores better gains that might occur in 2 steps.

A solution to this problem of feature grainsize, in which complex features are favored because they contain multiple sources of variability, might be to create synthetic linear feature groupings that can be chosen as an ensemble for addition to the model with each step. This was suggested by our results which showed logsuc and logfail being retained for the student for the BestLR start as discussed above. Future work will allow some "features" to actually test a combination of features for a component. Such a fix will allow us to add the combination feature logsuc and logfail (e.g., for the students) using 2 coefficients as usual, but in one stepwise step. This will allow it to compete with other terms such as logitdec or propdec, which incorporate success and failure already in the reported version here. More advanced methods can use factor selection which might be applied in both stepwise and LASSO within LKT, such as grouping specific features together such as KC models [21].

While the work here used BIC to reduce model complexity, and BIC works similarly to cross-validation in constraining unjustifiable complexity. We plan to confirm our results with out of sample validation methods in future work, also allowing us to further confirm that BIC is adequate. However, BIC likely underfits our final models relative to cross-validation [20]. So, it seems rather implausible that our models are invalid due to overfitting. Rather we may be running the risk of too little complexity, leaving explainable variability out of our model. Certainly, this highlights that out BIC stepwise threshold for addition and subtraction of terms was chosen arbitrarily to allow for interpretable models. We were pleased to see they also fit well.

4.1.1 Presets

To make the process of logistic regression modeling efficient, yet still retain some flexibility and user control, our tool includes a number of preset feature palettes that users will have available instead of specifying their own list. These presets are essentially a collection of theoretical hypotheses about the nature of the student model, given some goal of the modeler. The presets include the following four presets.

- Static - This present will contains only the intercept feature. It allows for neither dynamic nor adaptive solutions, essentially finding the best IRT type model, unless the item or KC component is not used, in which case it could simply find a single intercept for each student. Essentially it fits the LLTM model [5].
- AFM variants (i.e. dynamic but not adaptive) – This preset fits linear and log versions of the additive factors model[2], including LLTM terms that represent different learning rates or difficulties based on KC groupings (using the $ operator in LKT syntax).
- PFA and BestLR variants (dynamic and adaptive but recency insensitive) – This preset contains all of the above mechanisms, and also included the success and failure linear and log growth terms used in PFA [14] and BestLR[8].
- Simple adaptive – This catchall preset will include rPFA[7] inspired terms such as logitdec and propdec, described in the is paper and elsewhere [15]. In addition it will include temporal recency functions using only a single non-linear parameter, the best example of which, recency, was described in this paper and has been previously described [15].

Finally, future work might explore how Lasso also offers a convenient opportunity to evaluate the learner and KC model simultaneously. Within the Lasso approach, the coefficient of each KC can be pushed to zero and this could be used to allow refinement of the KC model. A limitation of our work here is that we did not explore this further, merely observing that in the models only some KCs were being assigned intercepts.

5. CONCLUSION

We find that the first few selected features in most models produced by the stepwise procedure are both effective AND interpretable. Articulating a theory to describe the simple models is relatively easy, since each feature can be justified by some research-based argument. For example, we see the importance of tracing student level individual differences in all the models, and we see the recency feature as indicating forgetting occurs. The LASSO procedure largely confirms the stepwise models are not far from a more globally optimal solution for our test cases and may reveal the future of the endeavor because of higher likelihood of a more global solution with LASSO despite the somewhat less interpretable models.

The present work sought to simplify the learner model building process by creating a model building tool, released as part of the LKT R package. Our promising interim results demonstrate too modes our tool has available to build models automatically. With stepwise, they can start with an empty model, provide sample data, and the fitting process will provide a reasonable model with a reduced set of features according to a preset criterion for fit statistic change. Alternatively, with the LASSO approach, the user provides data, and the resulting output will be a set of possible models of varying complexity based on a range of lambda penalties. The tool highlights models from lambda values based on minimum BIC and inflection points like those depicted in Figures 1 and 2.

6. ACKNOWLEDGMENTS

This work was partially supported by the National Science Foundation Learner Data Institute (NSF #1934745) project and a grant from the Institute of Education Sciences (ED #R305A190448).

7. REFERENCES

[1] Cen, H., Koedinger, K., and Junker, B., 2006. Learning Factors Analysis: A General Method for Cognitive Model Evaluation and Improvement. In *International Conference on Intelligent Tutoring Systems*, M. Ikeda, K.D. Ashley and T.-W. Chan Eds. Springer, Jhongli, Taiwan, 164-176.

[2] Cen, H., Koedinger, K.R., and Junker, B., 2008. Comparing two IRT models for conjunctive skills. In *Proceedings of the Proceedings of the 9th International Conference on Intelligent Tutoring Systems* (Montreal, Canada2008), 796-798.

[3] Chi, M., Koedinger, K.R., Gordon, G., Jordan, P., and VanLehn, K., 2011. Instructional factors analysis: A cognitive model for multiple instructional interventions. In *Proceedings of the 4th International Conference on Educational Data Mining (EDM 2011)*, M. Pechenizkiy, T. Calders, C. Conati, S. Ventura, C. Romero and J. Stamper Eds., Eindhoven, The Netherlands, 61-70. DOI= http://dx.doi.org/http://doi:10.1.1.230.9907.

[4] Conati, C., Porayska-Pomsta, K., and Mavrikis, M., 2018. AI in Education needs interpretable machine learning: Lessons from Open Learner Modelling. *arXiv preprint arXiv:1807.00154*.

[5] Fischer, G.H., 1973. The linear logistic test model as an instrument in educational research. *Acta Psychologica 37*, 6 (1973), 359-374. DOI= http://dx.doi.org/http://doi:10.1016/0001-6918(73)90003-6.

[6] Friedman, J., Hastie, T., and Tibshirani, R., 2010. Regularization Paths for Generalized Linear Models via Coordinate Descent. *J Stat Softw 33*, 1, 1-22.

[7] Galyardt, A. and Goldin, I., 2015. Move your lamp post: Recent data reflects learner knowledge better than older data. *Journal of Educational Data Mining 7*, 2 (2015), 83-108. DOI= http://dx.doi.org/https://doi.org/10.5281/zenodo.3554671.

[8] Gervet, T., Koedinger, K., Schneider, J., and Mitchell, T., 2020. When is Deep Learning the Best Approach to Knowledge Tracing? *JEDM| Journal of Educational Data Mining 12*, 3 (2020), 31-54.

[9] Gong, Y., Beck, J.E., and Heffernan, N.T., 2011. How to construct more accurate student models: Comparing and optimizing knowledge tracing and performance factor analysis. *International Journal of Artificial Intelligence in Education 21*, 1 (2011), 27-46. DOI= http://dx.doi.org/http://doi:10.3233/JAI-2011-016.

[10] Khosravi, H., Shum, S.B., Chen, G., Conati, C., Tsai, Y.-S., Kay, J., Knight, S., Martinez-Maldonado, R., Sadiq, S., and Gašević, D., 2022. Explainable Artificial Intelligence in education. *Computers and Education: Artificial Intelligence 3*(2022/01/01/), 100074. DOI= http://dx.doi.org/https://doi.org/10.1016/j.caeai.2022.100074.

[11] Papousek, J., Pelánek, R., and Stanislav, V., 2014. Adaptive practice of facts in domains with varied prior knowledge. In *Proceedings of the 7th International Conference on Educational Data Mining (EDM 2014)*, J. Stamper, Z. Pardos, M. Mavrikis and B.M. Mclaren Eds., 6-13.

[12] Pavlik Jr, P.I., Eglington, L., and Zhang, L., 2021. Automatic Domain Model Creation and Improvement. In *Proceedings of The 14th International Conference on Educational Data Mining*, C. Lynch, A. Merceron, M. Desmarais and R. Nkambou Eds., 672-676.

[13] Pavlik Jr., P.I., Brawner, K.W., Olney, A., and Mitrovic, A., 2013. A Review of Learner Models Used in Intelligent Tutoring Systems In *Design Recommendations for Adaptive Intelligent Tutoring Systems: Learner Modeling*, R.A. Sottilare, A. Graesser, X. Hu and H.K. Holden Eds. Army Research Labs/ University of Memphis, 39-68.

[14] Pavlik Jr., P.I., Cen, H., and Koedinger, K.R., 2009. Performance factors analysis -- A new alternative to knowledge tracing. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, V. Dimitrova, R. Mizoguchi, B.D. Boulay and A. Graesser Eds., Brighton, England, 531–538. DOI= http://dx.doi.org/http://doi:10.3233/978-1-60750-028-5-531.

[15] Pavlik, P.I., Eglington, L.G., and Harrell-Williams, L.M., 2021. Logistic Knowledge Tracing: A Constrained Framework for Learner Modeling. *IEEE Transactions on Learning Technologies 14*, 5, 624-639. DOI= http://dx.doi.org/10.1109/TLT.2021.3128569.

[16] Pelánek, R., 2016. Applications of the Elo rating system in adaptive educational systems. *Computers & Education 98*(2016/07/01/), 169-179. DOI= http://dx.doi.org/https://doi.org/10.1016/j.compedu.2016.03.017.

[17] Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L.J., and Sohl-Dickstein, J., 2015. Deep Knowledge Tracing. In *Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS 2015)* (2015), 1-9.

[18] Schmucker, R., Wang, J., Hu, S., and Mitchell, T., 2022. Assessing the Performance of Online Students - New Data, New Approaches, Improved Accuracy. *Journal of Educational Data Mining 14*, 1 (06/24/2022), 1-45. DOI= http://dx.doi.org/10.5281/zenodo.6450190.

[19] Smith, G., 2018. Step away from stepwise. *Journal of Big Data 5*, 1 (2018/09/15), 32. DOI= http://dx.doi.org/10.1186/s40537-018-0143-6.

[20] Yates, L.A., Richards, S.A., and Brook, B.W., 2021. Parsimonious model selection using information theory: a modified selection rule. *Ecology 102*, 10 (2021/10/01), e03475. DOI= http://dx.doi.org/https://doi.org/10.1002/ecy.3475.

[21] Yuan, M. and Lin, Y., 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68*, 1 (2006/02/01), 49-67. DOI= http://dx.doi.org/https://doi.org/10.1111/j.1467-9868.2005.00532.x.

[22] Yudelson, M., Pavlik Jr., P.I., and Koedinger, K.R., 2011. User Modeling – A Notoriously Black Art. In *User Modeling, Adaption and Personalization*, J. Konstan, R. Conejo, J. Marzo and N. Oliver Eds. Springer Berlin / Heidelberg, 317-328. DOI= http://dx.doi.org/10.1007/978-3-642-22362-4_27.

KC-Finder: Automated Knowledge Component Discovery for Programming Problems

Yang Shi[1], Robin Schmucker[2], Min Chi[1], Tiffany Barnes[1], Thomas Price[1]
[1]North Carolina State University
[2]Carnegie Mellon University
yshi26@ncsu.edu, rschmuck@andrew.cmu.edu, {mchi, tmbarnes, twprice}@ncsu.edu

ABSTRACT

Knowledge components (KCs) have many applications. In computing education, knowing the demonstration of specific KCs has been challenging. This paper introduces an entirely data-driven approach for (i) discovering KCs and (ii) demonstrating KCs, using students' actual code submissions. Our system is based on two expected properties of KCs: (i) generate learning curves following the power law of practice, and (ii) are predictive of response correctness. We train a neural architecture (named KC-Finder) that classifies the correctness of student code submissions and captures problem-KC relationships. Our evaluation on data from 351 students in an introductory Java course shows that the learned KCs can generate reasonable learning curves and predict code submission correctness. At the same time, some KCs can be interpreted to identify programming skills. We compare the learning curves described by our model to four baselines, showing that (i) identifying KCs with naive methods is a difficult task and (ii) our learning curves exhibit a substantially better curve fit. Our work represents a first step in solving the data-driven KC discovery problem in computing education.

Keywords

Computing Education, Knowledge Component, Interpretable Deep Learning, Neural Network, Code Analysis, Learning Representation

1. INTRODUCTION

Modeling new learning domains is a common task for researchers and educators [44, 23] which often involves identifying a set of domain-relevant skills[1], and determining when (e.g., in which problems) students practice each individual skill. Multiple data-driven approaches have been proposed to either improve existing learning domain models or to discover fully new models automatically using student log data. The benefit of such approaches is that they can alleviate the need for expert authoring (e.g., via cognitive task analysis [11]), and can reveal skills and problem relationships that may be counter-intuitive to the domain experts (e.g., due to blind spots [32]). Traditionally, these methods (e.g. [7, 9, 38, 24, 8, 15, 25, 34, 33]) output a Q-matrix, which maps the discovered skills to individual practice problems. With a defined Q-matrix, researchers and educators can leverage student modeling techniques such as knowledge tracing [13] to assess what exact skills the student knows and doesn't know and can provide personalized problems recommendations that target the student's specific knowledge gaps [7].

However, in domains such as programming where there exists heterogeneity in viable solution paths, simply knowing which skills are relevant to a given problem is often insufficient – we also want to know *when each of those individual skills is successfully demonstrated* in student practice, and when it is not. For example, if a student attempts a problem that requires the use of multiple skills (e.g. conditionals, iteration, logic, etc.), it is helpful to know which of these skills they have demonstrated successfully. This can aid us (i) better understand how students struggle and learn, and (ii) adapt teaching to individual student's needs by offering personalized help and instruction. In other words, rather than viewing success on a problem as a binary outcome (correct/incorrect), it would be helpful if a model could detect how successful a student's attempt is along the dimensions of each of the problem-relevant skills (e.g., loop correctness, iteration correctness, etc.). In domains like programming, many problems require students to apply multiple skills, and it is difficult to break problems down into single-skill substeps. Doing so requires making use not only of binary correctness information, as in prior work [9, 24, 38], but also information from students' actual code submissions. Note that this goal is distinct from that of predictive student modeling (i.e., knowledge tracing [13]), which in programming tasks predicts students' binary submission correctness (such has been done in [49]); instead, we are concerned with detecting more *fine-grained evidence* of knowledge being demonstrated during practice (i.e. successful or unsuccessful demonstration of multiple skills, rather than a whole problem).

As a first step towards this goal, this work explores how well a data-driven approach can discover *candidate* KCs (skills) that (i) can be detected automatically from student code

[1] Skills and knowledge components are used interchangeably in this paper. Knowledge components (KCs) are defined in [23] and introduced in Section 2.1.

Y. Shi, R. Schmucker, M. Chi, T. Barnes, and T. Price. Kc-finder: Automated knowledge component discovery for programming problems. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 28–39, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115671

submissions – allowing us to track when students successfully demonstrate each of them, and that (ii) conform to the learning theoretic properties of KCs suggested in prior literature [23]. For (ii) specifically, we attempt to discover candidate KCs that **fit idealized learning curves** [53] – meaning that students get better as they practice individual skills, quickly at first and then more slowly. The student error rate reduction over time is assumed to follow a power law (namely the power law of practice) [53]. KCs are also expected to be informative for predictions of students' success on the current problem [13]. Our goal in this work is to first understand how well the discovered candidate KCs meet these criteria, and then to explore how they can inform our understanding of student learning. We propose the KC-Finder algorithm which takes as input sequence data describing students' code submissions on a series of practice problems, and that outputs: (i) a set of candidate KCs; (ii) a Q-matrix mapping these KCs to individual practice problems where they are relevant; and (iii) a detector that can estimate, for a given student attempt on a problem, confidence values describing which of the relevant KCs were demonstrated correctly, allowing us to reason about why an incorrect attempt was made. We introduce a loss function inspired by learning curve analysis to train a deep learning model whose predictions conform to idealized learning curve [9]. We evaluate our approach by answering two research questions (RQs) using data from 351 undergraduate students in an introductory Java course:

- **RQ1:** To what degree do the discovered candidate KCs conform to the learning theoretical properties of KCs?
- **RQ2:** What kind of patterns have we discovered as KCs in students' code and how do they inform our understanding of student learning?

For **RQ1** we evaluate the candidate KCs by calculating a loss describing the fit to expected learning curves. For **RQ2**, we conduct a case study analysing concepts and skills tracked in student submissions. Our findings suggest that the KCs discovered by our data-driven approach induce learning curves conforming to the power law of practice. The discovered KCs are sometimes (but not always) meaningful and non-obvious to domain experts. However, we also found that the discovered KCs were no more predictive of student success than random, laying the groundwork to explore how to satisfy both learning theory and predictive performance.

2. RELATED WORK

2.1 Knowledge Components

We use "knowledge component" (KC) as a term to describe the skills students learn by practicing a set of programming problems in the computing education domain. The concept of KCs was introduced in the Knowledge-Learning-Instruction (KLI) framework by Koedinger et al. [23]. The KLI framework connects teaching and assessment via observable and unobservable events in the student learning process: instructional events, assessment events, and learning events. Learning events are defined as cognitive (or from a biological view, brain) changes occurring when students learn. While learning events cannot be directly observed or controlled, they are caused by instructional events such as explanations and lectures which are observable. Assessment events (exams, discussions, etc.) are used to probe the student's knowledge state which on its own is not directly observable. The framework defines the knowledge students learn through unobservable learning events as KCs which are a concept that builds the bridge between learning events and assessment events. In the general KLI framework, KCs can also refer to other terms (e.g. concept, principle, or fact). In this specific paper, we refer to KCs as skills, and by knowing a skill, we mean knowing certain concepts/principles/facts and how/when to use them. While there may be different kinds of skills (procedural, declarative, etc.), we do not distinguish these since this paper's focus is to find any skills relevant to programming tasks regardless of their nature. KCs can have different levels of granularity: for example, in programming, "knowing how to write iterations" is a skill, however, a more fine-granular KC can be "knowing how to use `for` correctly". Problem-KC relationships enable us to track students' knowledge mastery as they work through a set of problems [1] via knowledge tracing algorithms. Well-defined KCs and problem-KC relationships are essential for knowledge tracing algorithms such as Bayesian knowledge tracing (BKT) [13], AFM [9], PFA [37] and DKT [39] which estimate a student's mastery of skills based on the correctness of their responses to previous practice questions. The modeling of student knowledge states enables intelligent tutoring systems (ITSs) to adapt the workflow to individual students. For example, SE-COACH [12] uses KC-driven models to decide steps that need explanations, and Salden et al. [45] used KC-based student models to examine the process of studying worked examples and how knowledge is transfered when solving problems. In our work, KC-Finder automatically discovers candidate KCs from student code submissions for these systems to work in the CS education discipline.

2.2 KC Discovery & Data-Driven Refinement

Many student modeling tasks require the definition of KCs and problem-KC relationships. The task of identifying a set of suitable KCs and assigning them to individual practice problems is complex and requires substantial effort from domain experts and techniques such as cognitive task analysis (CTA) [11]. Even then the resulting KCs can suffer from biases and blind spot effect [32] inducing a need for additional refinement techniques (e.g., [7, 9, 16]). Further, the design of detectors that determine when a KC is demonstrated when a student attempts a certain practice problem is highly labor-intensive [24]. Data-driven techniques that leverage student log data have been proposed to refine existing expert Q-matrices (e.g., [7, 9, 24, 16, 34]) and to discover new KCs (e.g., [38, 8, 25, 34, 33]). These approaches demand less effort from human experts and can mitigate blind spot effects, but they may lead to less interpretable KCs.

One common method to evaluate KCs is learning curve (LC) analysis [9]. When evaluating KCs one hypothesis underlying LC analysis is that the collective error rate of a population of students in a KC decreases as they practice more. This trend is assumed to follow an exponential curve (e.g., the power law of practice [53]). Multiple methods have been proposed to improve the domain-specific Q-matrix using LC analysis. For example, learning factors analysis (LFA) [9] combines the additive factors model (AFM) with A* search to refine an expert Q-matrix, and relies on learning curve fit as optimization criterion. In this paper, our idea is similar to LFA. We also use learning curves to guide the optimization process, but differences exist. The biggest difference is that we do not require initially defined KCs and Q-matrix, and our main output is a set of discovered KCs. In addition to using learning curves in our loss function and evaluation metrics, inspired by performance predictions approaches (such as BKT [13] and DKT [39], and newer models that use learning curves in knowledge tracing [55]), we also add response correctness and actual student code submission information into the model optimization process to discover KCs suitable for performance predictions. This is similar to the work by Shi et al. [49] who incorporated code information into DKT. Shi et al. [49] also worked on a deep learning model, but the key difference is that we propose to discover KCs, while they focus solely on predicting student performance.

2.3 Student Modeling in CS Education

Student modeling in computer science education has its own challenges which set it apart from other domains such as math and science education. For example, open programming problems are hard to perform knowledge tracing on due to the inherent complexity of the individual problems and the heterogeneity of viable solutions. Some prior works aimed at finding KCs suitable for CS education. While the Lisp tutor [3] introduces KCs with the tutor's design, it does not allow students to write code freely, which greatly constrains the space of possible student code submission. It is difficult for teachers to perform CTA for open coding problems and to find suitable KCs. Gusukuma et al. [21] proposed a framework to identify misconceptions and find KCs accordingly, but the approach still requires substantial effort from domain experts, and the KCs they discovered have not yet been evaluated quantitatively. Rivers et al. [44] proposed using normalized nodes from abstract syntax tree (AST) representations of student code as KCs, and evaluated them with learning curve analysis. However, some KCs discovered by their model did result in learning curves not conforming to the power law of practice. One can hypothesize that this might be due to limitations of canonicalization algorithms (the process of converting student code into a standardized format). We look at this problem from a different point of view and hypothesize that the guidance of learning curves in the KC discovery process can help recover better fitting learning curves. We use a neural network structure subjected to a constraint inspired by ideal learning curves to identify KCs that warrant well-fitting representations. There is also related work with focus on applications of student models in CS education. Yudelson et al. [56] extracted student code features and used them for code recommendation; finding KCs can also help us attribute errors from student code, and thus may aid in subgoal detection tasks [31] and may enable better feedback [40] and hints [43] to students. Some recent works focused on student performance prediction [30, 28, 22, 58] by leveraging code submissions (though using experts or data-driven code features). Finding suitable KCs may help such models make more accurate predictions. Discovering KCs is still a key mission in CS education to improve many of these applications.

2.4 Deep Code Learning for CS Education

The advancement of deep learning and big data analysis algorithms has significant impacts in diverse domains including code analysis [2, 57, 10]. Many related techniques have found application in the CS education domain due to the increasing size of available datasets [26]. A frequently applied model is code2vec [2], which has been used to detect bugs and misconceptions from student code [50, 52, 51], and recent extensions also approach student performance prediction tasks [49, 29]. Other research used code2vec for general classifications of educational code in a block-based setting [19]. The recent Codex model (and the related CoPilot tool [10]) caught the attention of many CS educators. Codex is widely used for code auto-generation tasks, and has also achieved promising resulting when used to generate student code explanations [46]. While deep learning-based approaches often yield high predictive performance, they tend to be less interpretable then traditional modeling approaches (despite the effort from [17]) and it is difficulty to extract insights into the learning process that can be explained to students and teachers. Our approach leverages learning curve analysis to guide the model training process and aims to build a more trustworthy and explainable deep learning model for student modeling tasks in CS education.

3. METHOD

Our target is to discover KC candidates using constraints inspired by learning theory. Suitable KCs are expected to be informative in student performance predictions and should induce learning curves that follow the power law of practice. Overall, there are four assumptions about KCs made in our work to build the KC discovery model. They are introduced below along with the theoretical rationales behind them.

3.1 Assumptions

The assumptions underlying the model design are **A1:** The collective error rate of students on a given KC *decreases* with subsequent opportunities to practice that KC. This decreasing trend is assumed to follow the power law of practice [9, 53]. **A2:** The demonstration of KCs in a problem solution attempt should be *predictive* of the attempt's correctness. **A3:** KCs are *detectable* from a student's current code submission. **A4:** All problems have a *fixed set of KCs*, meaning that the related KCs for a problem are fixed, independent of the submissions from students. **A5:** All KCs have the same *initial error rate and learning rate.*

A1 states that observations from students practicing KCs should induce learning curves that conform to the power law of practice. It is natural to assume that when practicing KCs, as students practice more, they become more proficient in these KCs, and thus make fewer mistakes. The power law of practice postulates that the collective error rate from all students decreases as students practice more following a power function ($Y = aX^{-b}$). This assumption has been made by multiple prior student modeling approaches [9, 44].

A2 assumes that if a student knows all KCs relevant to a problem, it indicates they are likely to answer the problem correctly. When incorporating this assumption into a data-driven model, it implies that the demonstration of KCs relevant to a problem in a solution attempt should be predictive of attempt correctness. For example, if a problem requires the application of KCs A and B, a successful demonstration of KC A should suggest an increased likelihood of getting the problem right. This assumption has been used in many knowledge tracing models [13, 6, 35, 47] that make predictions on future submissions with a focus on domains with closed and structured questions. However, for open-ended, free-form computer science problems, the complexity and intertwinedness of individual KCs may cause more difficulty in performance prediction. We thus limit our assumption to the current submission correctness.

A3 postulates that KCs are observable and detectable from students' code submissions. We only have access to the code submissions, and in this work we only focus on KCs that can be extracted from code submissions. While some KCs may exist that are not directly observable through code submissions (for example, reading skills are also required when students try to solve programming tasks specified by text requirements), various KCs can be observed in code submissions. For example, Rivers et al. [44] extracted various KCs represented by AST nodes derived from code submissions.

A4 and **A5** are assumptions specifically made in the design process of our current model. They may not necessarily be true, but we use them to facilitate the creation and training of the KC-Finder model. Future work can focus on loosening these assumptions. A4 as a limitation, states that the number of practiced KCs is fixed for each problem. In many cases this is true, but whether students practice certain KCs can also be affected by the code they write and the solution path they choose. For example, in an open-ended programming problem, the instructor may expect students to solve the problem by using nested `if` conditions, but some students may choose more complex logic operations to avoid nested `if`. Under this assumption, we assume that the student *did not demonstrate* a correct practice of the required KC "nested `if`", while the student actually indeed *correctly practiced* the KC "complex logic operations" and "nested `if`" is **not required** by the problem. We use this assumption to allow the usage of the Q-matrix since if KCs are dependent on submissions, one Q-matrix cannot represent the KC-problem relationship since the relationship varies across different submissions. In A5, we have an assumption that requires all candidate KCs we discover to have the same starting error rate (a) and learning rate (b) in the power function for their learning curves. We set this assumption as a start for using properties of the power law curve, and save the automatic fitting of more specific learning curve parameters for later work. Our experiments show that we can still discover meaningful KCs under this assumption.

3.2 KC-Finder Model

Under the guidance of these theoretical properties, we define the KC-Finder model structure below. Figure 1 provides an overview of the model. We show a single student and their T code submissions $\{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_T\}$ for a course as example to illustrate the process of the model. The output of the model is specified in orange in the figure, where the current submission correctness is indicated as $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_T\}$.

In a submission $\mathcal{S}_t$, two types of information are available to the model for the correctness classification task: the problem ID p_t and the actual code submission c_t. While both cannot be immediately processed by a mathematical model, we separately represent the problem ID and code submission information as real-valued vectors. For the problem ID, we use one-hot vectors to represent the IDs as vectors, using a vector $\mathbf{x}_t$ to represent p_t, where the length of the vector equals the number of problems in the dataset, with all elements assigned as 0, except the element associated with the problem assigned as 1. Note that this setting is similar but different to typical knowledge tracing tasks [39, 49, 4] which also use one-hot representations for problem IDs. Knowledge tracing includes the correctness information in the task of the *next* submission performance prediction (which is denoted as $\mathbf{y}_{t+1}$). In contrast, we do not include the correctness information since our task is to classify the correctness of the *current* submission $\mathbf{y}_t$, and to *discover KCs* through the model learned from this task. Code submissions c_t are embedded through a code2vec model [2], which has been recently introduced to educational analytics in multiple tasks [51, 19]. The code2vec model can embed a code snippet c_t into a vector $\mathbf{z}_t$. This part of the model updates parameters in the training process, along with other layers in the model.

Neural networks have a common structure. Linear layers (also called fully-connected layers) are defined by weight matrices and apply linear transformations to vector inputs. The product of these multiplications is often followed by non-linear functions such as sigmoid or tanh to introduce non-linearity into the model. In the KC-Finder model, all linear layers ($\mathbf{W}_{KC}$, $\mathbf{W}_c$ and $\mathbf{W}_p$) have the same mathematical operations (with different weights), which first multiply with the input vectors and then apply the sigmoid function (denoted as $\phi(\cdot)$) to every element to compute the output. For example, when a code vector $\mathbf{z}_t$ passes through the linear layer $\mathbf{W}_c$, the equation for this process is $\mathbf{h}_t = \phi(\mathbf{W}_c \mathbf{z}_t)$.

The vector $\mathbf{h}_t$ is of dimension L, where L is the total number of KCs. We intend to interpret $\mathbf{h}_t$ as the error rate of students practicing KCs $\{l = 1, 2, ...L\}$, but there are some challenges. First, not all problems practice all KCs, and our model needs to learn problem-KC relationships. To this end, we leverage the one-hot problem embeddings $\mathbf{x}_t$ to infer weights and represent the KCs corresponding to the current problem. We use linear layer $\mathbf{W}_{KC}$ to learn a relationship between potential KCs and problems, and use a sigmoid function to scale the output of layer $\mathbf{m}$ to the $[0, 1]$ range. The layer-$\mathbf{m}$ weights then multiply with the values of $\mathbf{h}_t$ and generate a vector $\mathbf{k}_t$. The intuition behind this is that every problem can have a probability of practicing certain KCs, and we use $\mathbf{m}$ as this probability and multiply the $\mathbf{h}_t$ vector to represent the selected KC values. The output $\mathbf{k}_t$ is a masked representation of the knowledge status of a student. When using an ideal learning curve to force the distribution of the representations across students in a batch, it can readily be interpreted as the error rate of students practicing KCs as we expect it to follow the power law of practice. For a batch of students' submissions that practiced a KC, cumulatively they should also follow the power law of practice, and preserve their ability to predict the submission's correctness. These two properties lead to the design of the loss function which is used to train the KC-Finder model:

$$\mathcal{L} = \alpha(\frac{1}{N}\sum_{N} H(\mathbf{y}, \hat{\mathbf{y}})) + (1-\alpha)(\sum_{T,L} |\frac{1}{N}\sum_{N} k_{n,t,l} - \hat{k}_{t,l}|) + \gamma(||\mathbf{W}_{KC}||_1). \tag{1}$$

In Equation 1, the loss when the model has a batch of N student submissions comes from three sources. The first part

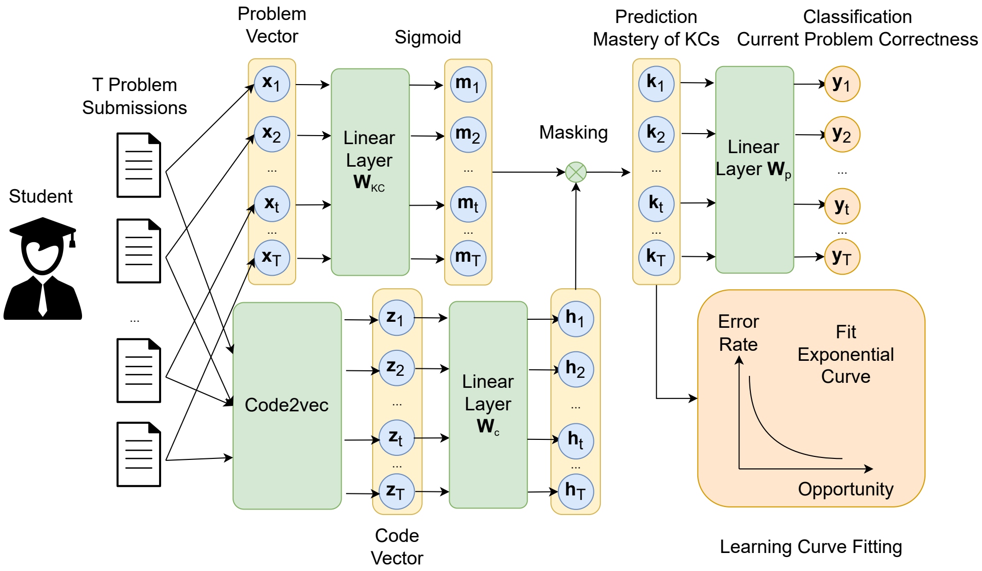


Figure 1: KC-Finder Model structure, where blue nodes are vectors, and green blocks represent neural network structures.

$H(\cdot)$ is the binary cross-entropy [14] of the classified results y and the ground truth correctness $\hat{y}$, leading the model to learn weights that produce a low submission correctness prediction error. The second part is the loss for the fitness of the learning curve to encourage predictions that conform to the power law of practice. Since the learning curve is calculated through a batch of students, we first average the error rate of practiced KCs for every student in the batch. For a certain knowledge component l, we calculate the assumed learning curve, where $\hat{k}_t = at^{-b}$. In the equation, a stands for the starting error rate for students when they have not practiced a skill, while b denotes the learning rate. Because we cannot estimate KC-specific a, b parameters a priori without employing additional information or assumptions, we assume all KCs have the same a, b parameters. This learning curve fitness loss is optimized so that the model produces $\mathbf{k}$ vectors that can be interpreted as the error rate of practiced KCs. We use an α hyperparameter to control the importance of the classification loss and the fitness loss. The last part of the loss is an L1-norm regularization term weighted by hyperparameter γ which ensures sparsity in the $\mathbf{W}_{KC}$ weights and thus creates a more sparse masks $\mathbf{m}$, allowing KCs to be removed from unrelated problems. While the output of this classification is a binary label describing whether a student succeeds in their submission, the key product of the work is the Q-matrix specified in $\mathbf{m}$ and the learning curves $\mathbf{k}$ on the corresponding knowledge components.

4. EXPERIMENT

4.1 Dataset

Our experiments use the publicly available CodeWorkout dataset[2]. The dataset is collected from an introductory Java course at Virginia Tech in Spring 2019. The dataset is stored in ProgSnap2 [42] format and is released to the public in the 2nd CSEDM data challenge[3]. No identifiable information (such as geographical information, GPA, etc) on individual students is released, and the dataset has been anonymized for ethical considerations. The dataset includes submissions from 410 students for 50 programming problems, which are grouped into 5 assignments according to the topics. For example, the first assignment mainly focuses on the `if` conditions, while a later assignment has more problems on `for` loops. The typical length of the student code submissions is 10 to 20 lines and 41.86 tokens, submitted to the CodeWorkout [18] platform, and tested by pre-defined test cases. The unknown tokens are specifically assigned as a unique identifier [unk] in the model. To avoid overfitting to the problems, user-defined variables and strings are also normalized to a fixed string in students' code. On average, 23.68% of all submissions (from all students) are correct, meaning they passed all test cases. Our model involves training, validation, and testing phases. We split the dataset according to students by a ratio of 3:1:1 for each of the three phases. We train the model with training data, use the validation set to tune hyperparameters, and test and evaluate the model on the testing set. The results are averaged through 5 times repeated sampling to ensure the result are reliable.

4.2 Data Preprocessing

Code submissions are complex, and we only use the first attempts of students on each problem for potential KC discovery. One reason is that it has been common for knowledge tracing tasks to only consider the first submissions for problems [48, 4], as students practice skills when they first see the problem and have not received any feedback on the specific problems. Another reason is that for code submissions, students tend to debug on their later submissions. This process involves more complicated behavior, which may not be fully explainable by conventional knowledge component modeling. Sometimes students even get intimidated by problems in case of repeated incorrect submissions, only to click the submit button multiple times and thus make invalid submissions that do not show what they know and don't know.

[2] https://pslcdatashop.web.cmu.edu/Files?datasetId=3458
[3] https://sites.google.com/ncsu.edu/csedm-dc-2021/home

Therefore, as it would be a non-trivial task to evaluate candidate KCs for multiple submission situations, we only use the first submissions from students for every problem. This is also different from other analyses such as “learning trajectory analysis” [54], as we only focus on the submissions when students practice skills for the first time.

Some students may also have exhibit cheating behavior in the dataset (since the system used to collect data don’t have a detector for cheating, unfortunately). We also observed students submitting partial code as their first submission or potential cheating. For example, some students struggled with easier problems, but suddenly are able to make correct submissions on their first try after a certain problem, and those later problems are generally more complicated than the ones they failed many times before.

We added preliminary filters to keep only the first attempt submissions from students and avoid students with possible cheating behavior. To only keep the first submissions, we implemented a filter to detect the first submissions that are not too short (longer than 3 lines of code) and only kept these first submissions. We also implemented a filter to remove students with sudden changes in their submission patterns. Out of 410, we finally kept 351 students in the dataset for the KC discovery task.

4.3 Hyperparameter Tuning

We used validation sets for hyperparameter tuning. Specifically, we applied grid search to find hyperparameters yielding the highest classification AUC scores on validation sets for model evaluations. This process is repeated 5 times to reduce the risk of overfitting. We selected the learning rate for the model as 0.005 through space of $0.001, 0.003, 0.005$, and the training epochs are selected as 80 through the early-stopping process. We also tuned the α parameter (specified in Equation 1) and used a value of 0.97 (in a range of 0.03 through 0.97). At the same time, the lower weight on learning curve fitting loss does not have a significant effect on the potential KCs discovered (they still produce good learning curves even if set low). Still, a higher weight on classification loss is needed for the classification task. The γ parameter controls the speed of removing potential KCs from unrelated problems and is set to a low value of $3e-5$. The other model hyperparameters are kept the same as the default ones in the settings of the code2vec and DKT models [2, 39]. We did not tune on the different combinations of the a and b parameters of the expected learning curves of the KCs since it is hard to evaluate the quality of potential KCs quantitatively, as in real applications, it is also feasible to try different a and b parameters and check the quality of potential KCs found under various combinations. In our experiments, we used the values $a = 0.7, b = 0.6$. We also assumed that $L = 30$ potential KCs exist in the dataset, and while the model is able to reduce the number by learning a candidate KC not practiced in any problem, the number is an educated guess by the authors after checking through the problems. Future experiments can be introduced to evaluate more value combinations. We kept these values since there is no direct way to quantitatively evaluate the quality of programming skills or misconceptions discovered by the models. The model is trained with a computer equipped with an Nvidia GeForce RTX 2080 Ti GPU. A single run of the training takes less than 10 minutes, and inference of a single batch of students takes seconds of computation. The code is implemented in PyTorch [36] and publicly available[4].

4.4 Baseline KC Models

We compared the KC discovered by our model with four baseline methods. One can argue that topics of the problems can be extracted and assigned as the KCs in each question. As the first baseline in our experiment, one of the authors manually examined the problem requirements and code solutions, identified a set of 15 *Topic KCs*, and manually tagged each problem with a set of relevant KCs. The second baseline used an alternative way to extract KCs that examines the student code submissions and extracts the most frequently used code components (show up in more than 20% of all solutions) in correct submissions as KCs for certain problems. This is a simplified KC model compared to [44], as we do not have an automatic hint generator for Java compared to their work for Python programs. We implemented this method to extract the nodes from the AST representation of student code, and filtered nodes that show up in more than 20% of correct submissions as KCs required by the problem, resulting in 21 *Node KCs*. Textual and numerical leaf nodes of ASTs were removed from KCs since they vary among problems. Lastly, we considered two standard baselines from prior work [37] one which uses a single KC for all problems (i.e., general programming knowledge), and another that defines a separate KC for each problem (i.e., each problem is its own KC). We compared our discovered candidate KCs with the four baselines using the fitness errors of the induced learning curves.

5. RESULTS

5.1 Learning Curves

We first show three example learning curves generated from our model using the testing dataset shown in Figure 2. To evaluate the fitness of the learning curves of each of the potential KCs, we calculated an average absolute error $e = \frac{1}{T}\sum_T |k_t - \hat{k}_t|$ to compare the curves to the expected curves under the exponential curve $\hat{k}_t = at^{-b}$, where the a and b parameters are automatically fitted. Note that for each KC candidate, only problems practicing the KC are counted when calculating the error e. For example, the KC candidate #5 is practiced in almost all problems. The KCs shown in Figure 2 all have a relatively low error compared with the assumed exponential curve. KC candidate #5 has an error of 0.037, KC candidate #4 has an error of 0.026, and KC candidate #2 has an error of 0.032 All KCs candidates we extracted have a $e < 0.1$, and the mean error is 0.034, showing that the learning curves of the potential KCs are generally consistent to the learning curve, and follow the power law of practice. On the other hand, the baseline KCs do not create KCs that fit the expected exponential learning curve. We show four learning curves created from the baseline Topic KCs and the node KCs in Figure 3. The two learning curves on the left represent the error rate of the *submissions* when certain Topic KCs are practiced, and the right side shows learning curves when node KCs are practiced. We can clearly see that neither KC models on the

[4] Code Repository:
https://github.com/YangAzure/KC-finder

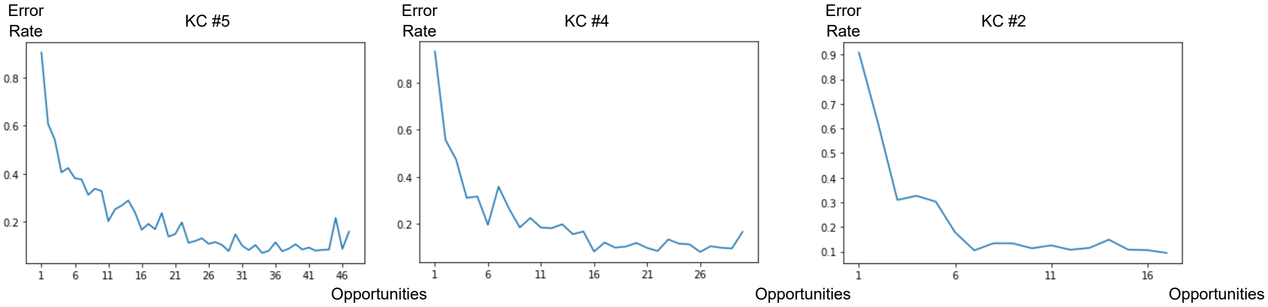


Figure 2: Learning curves of different KC candidates generated from students in test set.

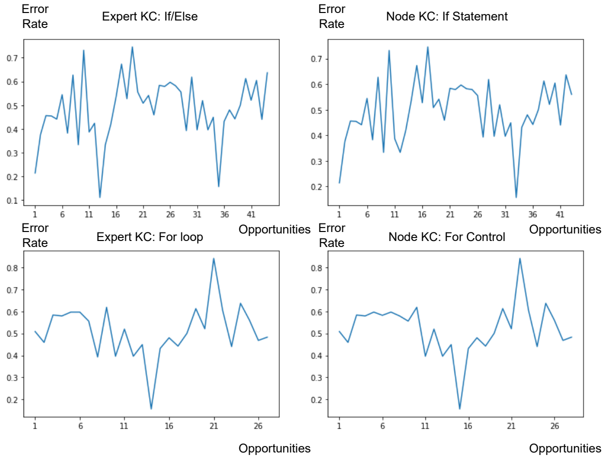


Figure 3: Learning curve examples of Topic KCs and Node KCs on concepts of `if` and `for` statements.

`if` or `for` statements show an exponential, or even decreasing trend in error rates. When calculating their fit to the assumed exponential learning curve for the model (with automatically fit parameters), we show in Table 1 that both KCs have very high fitness errors. Only a small subset of the KCs has a valid learning rate factor (b), presenting Topic KCs and node KCs cannot generate learning curves fitting the exponential curve with a fixed set of parameters. When looking at more learning curves generated from both KCs, they are similar to the examples in Figure 3 and do not have a decreasing trend over opportunities. The learning curves for Topic KCs and node KCs are similar to each other for a certain concept (for example the first two learning curves for `if` concept), which confirms that the expert extracted KCs are represented in code submissions as well. However, one limitation here is that the correctness may not directly represent the correct practice of certain KCs. The correctness metric represents students practicing all KCs in a certain problem correctly, but not on certain KCs. We use the correctness of the submissions to validate these two KC extraction methods as an approximation, which shows that our work can represent the learning curve for certain KCs without the need for a specifically designed partial evaluation of the submissions – only if we can explain the potential KCs discovered by the model.

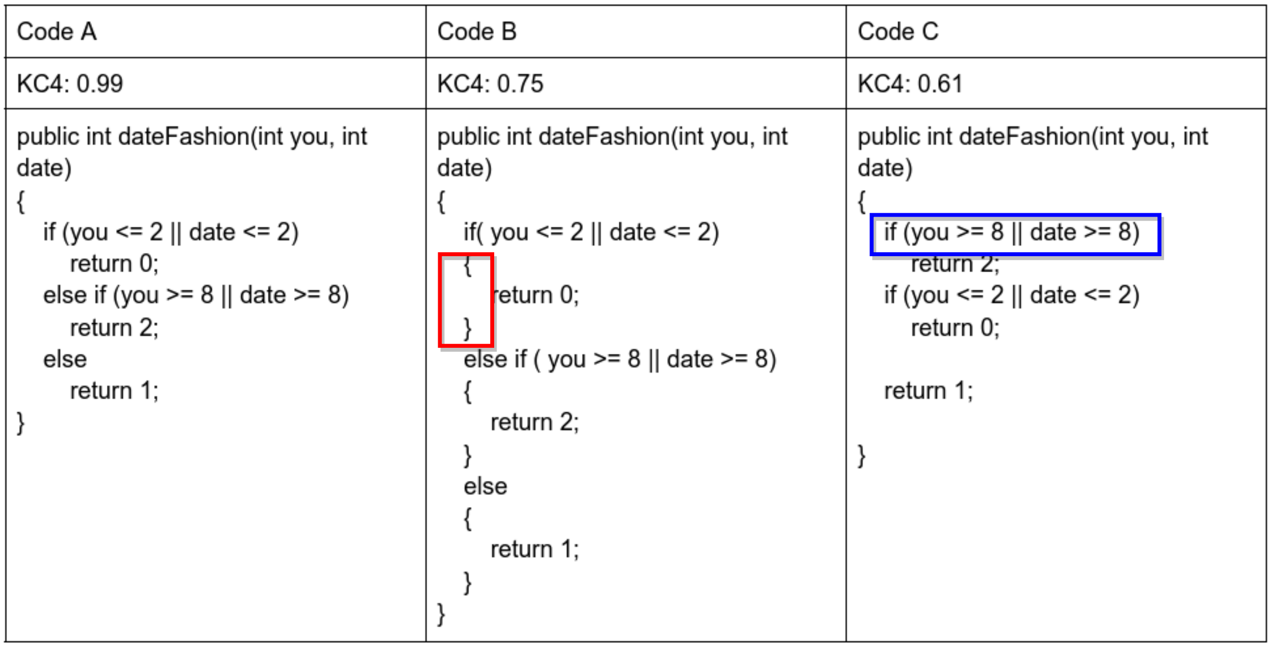
5.2 KC Interpretation

Our model discovers KC candidates represented as vectors that generate reasonable learning curves, and we show the interpretability of the KC candidates in this section. We manually examined the code and their corresponding KC values and found that we could find meaningful and interpretable KCs from these automatically discovered KC candidates. We manually inspected the discovered KCs across multiple problems, and show one example for the presentation purpose. In Figure 4, we show an example case of a KC candidate (KC #4) and explain what has been tracked in this KC in one problem. The problem requires students to use `if` conditions with logic operators, and one typical and non-obvious error from students is to use the order wrongly and return incorrect values that cause test cases to fail. It should be noted that the values do not indicate whether the KCs are practiced or not in certain problems. A low KC value means a failed demonstration of the candidate KC. Code A submission is correct, with a high KC value on KC candidate #4. Code C has a wrong order and has a low KC value on the same KC. While other reasons could cause the difference in KC values, Code C is incorrect due to the wrong order. This KC could possibly track bracket usage, as the only difference between Code A and Code B is bracket usage. Using brackets led to a lower KC value for Code B.

We also found some other concepts associated with KC candidate #4, for example, in one problem, all students who have an error in using `=` as comparison operator `==` got lower KC values. One KC may not represent a clearly defined concept by experts. The KC candidates are almost certainly amalgamations of different concepts, and no single behavior seems to explain the KC itself. Some of these concepts are consistent through different problems, as the example shows are all related to the if condition, and some of the concepts are problem-specific. As we do not have any specific design in the mode, KCs can be conceptual and can be distinct when they tend to improve together. Some candidate KCs are meaningful and important skills for the problem (e.g. the sequence of if conditions as shown in 4), which instructors might not have intuited; however

5.3 Code Classification

In our experiments, we report the classification results through 5 times running and calculate the average of the runs to get the classification results. KCs should be informative to predict the correctness of the code submission. To serve as a sanity check, before we evaluate the discovered potential



Code A	Code B	Code C
KC4: 0.99	KC4: 0.75	KC4: 0.61

Code A:

```
public int dateFashion(int you, int
date)
{
   if (you <= 2 || date <= 2)
      return 0;
   else if (you >= 8 || date >= 8)
      return 2;
   else
      return 1;
}
```

Code B:

```
public int dateFashion(int you, int
date)
{
   if( you <= 2 || date <= 2)
   {
      return 0;
   }
   else if ( you >= 8 || date >= 8)
   {
      return 2;
   }
   else
   {
      return 1;
   }
}
```

Code C:

```
public int dateFashion(int you, int
date)
{
   if (you >= 8 || date >= 8)
      return 2;
   if (you <= 2 || date <= 2)
      return 0;

   return 1;

}
```

Figure 4: Comparison of code submissions and their corresponding scores of KC #4. Frames show the possible code difference that triggered the KC value difference.

Table 1: Fitness error of Topic KCs, Node KCs, and KCs discovered by our model. The Valid LCs column indicates the number of LCs with a positive learning rate parameter ($b > 0$). A negative learning rate indicates a degenerate KC (students get worse with practice).

KC	Error	Valid LCs
Topic KC	0.0663	1
Node KC	0.0785	5
Model KC	**0.0342**	**30**

Table 2: AIC, BIC of Topic KCs, node KCs, one KC, all KCs, and KCs discovered by this model.

KC Model	AIC	BIC
One KC	3223.92	3651.62
All KC	3179.95	4189.79
Topic KC	3220.99	3672.45
Node KC	3214.84	3678.18
Random KC	3076.08	3848.31
Model KC	3081.44	3853.67

KCs, we evaluated an average of running the model on different splits of training and validation sets five times and reached an average AUC score of 77.26%. Considering that no correctness information is given to the model, and only discovered KCs are used for making the classification, the potential KCs can be used to classify the code correctness, showing the necessity of using the correctness information in the loss function.

5.4 Q-Matrix Analysis

We present the Q-matrix we found from the testing dataset, ranked by the number of problems in Figure 5, and compare the corresponding AIC and BIC scores in Table 2 to evaluate how well the discovered and baseline KCs predict student performance/correctness, following prior methods [9]. These metrics are frequently used to evaluate the goodness of fit. More detailed equations for the metrics can be found in prior works (e.g. [9]). The Q-matrix shows that the KCs are relatively evenly distributed through all problems with good sparsity. In the comparison results, the model scores are similar to a random KC model. It would be unsurprising to have this result, as the model learned KC candidates that were amalgamations of different and overlapping micro-concepts; therefore, it makes sense that a Q-matrix involving these KCs would not be meaningful. While our KC model does not generate better AIC/BIC scores than random KC, the other baseline models (even manually defined KC models). This shows that it is difficult to create a predictive KC model with the dataset. Furthermore, it suggests that fitting a learning curve itself is sufficient to discover KCs for domain modeling. We did not manually inspect the baselines' KC quality since they are specifically designed to represent different levels of KCs. For example, one KC and all KC are naive baselines that any domain could use, while the remaining baselines are expert-defined, and thus already fit to expert understanding of KCs. One direction for future work would be to seed the model with an expert-authored Q-matrix and allow the model to discover KC candidates which match the pre-specified pattern. This KC-refinement task has been explored by various works (e.g. [24]). However, doing so with our approach would help to address the challenge of figuring out which relevant KC a student is struggling with when they get a problem wrong.

6. DISCUSSION

6.1 Expected Properties

We first answer RQ1 in this discussion section: *How closely do the candidate KCs detect match the expected properties of KCs?*

Our goal was to discover candidate KCs that matched two important properties of high-quality KCs. First, students should be more successful at demonstrating KCs as they practice them, following the power law of practice [9, 24]. Second, a student's success on a given problem should be predicted by how successfully they demonstrated relevant KCs for that problem [13, 47]. To address this, we trained a model to detect candidate KCs and then evaluated those KCs on a separate dataset. Our findings suggest that the discovered KCs largely meet these two goals, much more so than the four baselines we compare with (see Section 4.4).

First, the resulting learning curves appear high quality, fitting well to a power law curve. By contrast, none of our baselines produced viable learning curves, suggesting that naive approaches for defining KCs are ineffective with our dataset. Similarly, prior work [44] has found that learning curves in programming often fail to align with learning curves. This result suggests that our model could detect patterns in student code that become more frequent as students practice – quickly at first and then slower with time – as suggested by the power law of practice [9]. As we discuss below, some of these patterns likely correspond to skills that students develop over time, such as the usage of `if` conditions, or (the absence of) misconceptions that become rare over time, such as using an assignment operator (`=`) instead of comparison (`==`) inside of a conditional statement. However, some of these patterns may simply correspond to code constructs that are used more frequently as the semester progresses (e.g. variables, which are rare in early assignments), and thus naturally increase in frequency, without in reality having much to do with practice. Matching learning curves is not itself enough to say a KC is meaningful, but it does suggest that some of the discovered KC candidates may correspond to learned skills.

Second, we found that, for a given problem, the relevant KC candidates were, collectively, predictive of students' correctness on programming practice problems (AUC = 77.26%). These results suggest that whether a student successfully demonstrates a candidate KC discovered by our model gives insight into whether they will succeed at the current problem. In other words, the code patterns underlying these candidate KCs are also important code patterns for solving the programming problems in our dataset.

Overall, these results suggest that the candidate KCs we discovered do match the expected properties of idealized KCs in these two dimensions. Importantly, the results we presented were from a hold-out test dataset with unseen students, suggesting that KC candidates can generalize across different students within a course. However, while these criteria are necessary, they are not sufficient, and we will explore the limitations of the discovered KCs below.

6.2 Skill Tracking

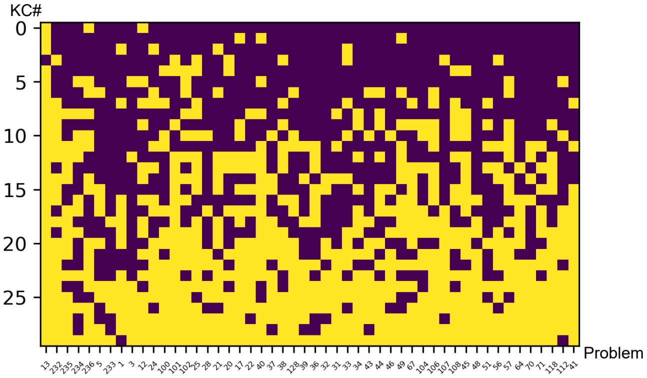


Figure 5: Q-Matrix representation of KCs and problems, where yellow cells represent a presence of the KCs in a problem, and dark cells represent an absence.

We answer RQ2 in this section: *What properties do the discovered KC candidates have? What kind of patterns have we discovered as KCs in students' code?*

First, we found that there are important differences between the discovered KC candidates and how experts would likely define KCs for a given domain. Rather than discrete concepts (conditionals), the KCs are amalgamations of different micro-concepts (e.g. correct ordering of the primary if-statements in a problem), where no single behavior seems to explain the KC itself, and different KCs overlap. Some of these micro-concepts show up across different problems (e.g. KC #4 detects the misuse of `=` in conditions (or `assignment in conditional`) misconception across various problems). Some of these are also problem-specific, e.g. the order of two if statements in the problem shown in 4. This suggests the need for further research on operationalizing the idea of a KC's "consistency" – that a KC should mean the same thing when detected across students and problems. This is a non-trivial idea to encode in a model, which lacks any domain expertise. Ideally, such a definition should be defined based on student behavior (e.g. if a KC is consistent, students' performance for problems that use the KC will be correlated). In some ways, this idea is operationalized by approaches such as AFM [9]. However, since the actual domain information is not used, it would not be feasible to evaluate the performance of such methods. Although the Q-matrix learned by the model was not meaningful, we also found that the candidate KCs can inform our understanding of what skills students develop in a domain. For example, we found that KC #4 clearly detected a micro-concept focused on students' use of brackets in code. Importantly, these brackets did not change the function of the student's code, and it is unlikely an expert would have thought to include them as a discrete skill. However, our model identified this pattern as being predictive of student success and fitting a good learning curve. In retrospect, this makes sense as skill students develop: acquiring familiarity with syntax and style conventions (e.g. when brackets are and are not necessarily) can help students succeed on various problems, even if it does not directly affect their correctness. We also found that candidate KCs included misconceptions, such as the confusion of `=` and `==` in `if` conditions. More work is needed to develop methods for extracting the meaning of these data-driven KC candidates, and use this understanding to develop insight.

6.3 Design Choices

We made design choices according to our assumptions (See details in Section 3.1). Assumption **A3** specified that KCs should be detectable from code, and thus we used the code2vec model to process code into vectors $\mathbf{z}$ [2]. Code2vec has been used for educational data mining tasks recently [50, 49, 19], and the code embedding extracting module can be other models such as ASTNN [29, 57] as well. Assumption **A4** specified that problems should have a fixed set of KCs, and we thus 1-hot embedded the problem IDs into vectors $\mathbf{x}$, and use them to calculate a set of masks $\mathbf{m}$ to specify the KCs active for different problems. The masks select relative KCs for processed code vectors $\mathbf{h}$, and the selected values $\mathbf{k}$ participate in the loss calculation. According to assumption **A1**, one requirement is that if we let $\mathbf{k}$ correspond to KCs, they should follow the power law of practice. We designed the loss so that $\mathbf{k}$ values fit an exponential curve, generated by a fixed set of parameters, relaxed by assumption **A5**. Finally, since **A2** specifies the performance of KCs should be predictive of the code correctness, we use $\mathbf{k}$ values to make the predictions and have the model also train on the classification loss. When we assume the learning curves have the same parameters, the model may overlook KCs with very different starting error rates and learning rates. We thus used an L1-regularization to encourage the sparsity of $\mathbf{W}_{KC}$ and allow KCs to drop. While many variations and improvements can be made, this model is a proof of concept and serves as a prototype for the KC discovery task in the programming education domain.

6.4 Research and Educational Implications

This work introduced a fully data-driven KC-discovery algorithm designed for the CS education domain. It uses student log data describing the correctness of student responses and actual student code submissions to discover KCs and map them to individual programming problems. It connects pieces of *computer science education* with *learning theories* to discover a Q-matrix which conforms to the power law of practice [53]. It has been a different task from the KC refinement methods such as LFA, which uses learning curve analysis for KC-refinement, but it requires an initial Q-matrix [9, 24]. These student model improvement methods can reduce the load on experts when performing cognitive tasks analysis [11]. In contrast, our model directly reduces expert effort by providing a student model that produces KCs with learning curves fitting the power law of practice. Our model leverages deep learning structures, typically known as "black boxes", however, we specifically designed the model such that the middle layer information can be interpreted as the KC ability estimates and thus made this model interpretable. The discovered KCs can be applied for performance prediction tasks by directly using the KC values or plugging the model structure to current knowledge tracing models for CS education [49]. While there are vector representations of student code submissions, they can also serve as language-agnostic representations to represent the mastery of KCs [27]. Our model can also be seen as a misconception attribution tool. When a student is predicted to have a high error probability on a certain KC, an automated hint or interference can be generated in an adaptive way to help the learning process [41]. Finally, the model can also be used to analyze the KCs covered by a set of problems using submission data from a semester, and thus to make more informed pedagogical decisions [20].

6.5 Limitations

Besides the assumptions we made to guide the model design, there are also other limitations present in this study. First, we did not incorporate the factor of using test cases. The run-time results of carefully designed test cases may contribute to the attribution of errors when students practice KCs. However, we do not have the full test case information for every problem in the dataset, and it is non-trivial to match KCs with specific test cases. Future work may integrate information about test case results into the existing method to enhance the KC discovery process. Second, we followed the tradition of knowledge tracing tasks and only used students' first submissions in model training and evaluation. One major limitation is that we cannot track the actual opportunities of practicing KCs in repeated submissions, especially if we don't assume that problems have fixed sets of KCs. We made this decision after the exploratory data analysis, during which we found lots of students made debugging submissions that are unnecessary. We found it can be complicated to explain this behavior, and are unsure if students actually intend to practice KCs when submitting a debugged code (for example, they may hit submission buttons multiple times, or they just wanted to exhaust possible choices to reach the correctness), as pointed out by Baker et al. [5]. It could also be interesting to investigate student behavior after their first submissions for programming problems. Finally, this model serves as a prototype and many variations could possibly generate better KCs. However, we do not have a metric to quantitatively evaluate the extent of KCs being reasonable and interpretable. While we do have a metric to examine the fitness of the learning curves and the classification of the code correctness, one limitation of the results is that the classification results are no better than random, and the discovered KCs are sometimes meaningful and non-obvious. The limited size of dataset may also cause a limited generalization on other datasets. We will explore ways to use expert knowledge to evaluate the quality of KC models, using this research to guide our future direction.

Acknowledgements

This material is based upon work supported by NSF under Grant No. #2013502 and #2112635.

7. REFERENCES

[1] V. Aleven and K. R. Koedinger. Knowledge component (kc) approaches to learner modeling. *Design Recommendations for Intelligent Tutoring Systems*, 1:165–182, 2013.

[2] U. Alon, M. Zilberstein, O. Levy, and E. Yahav. code2vec: Learning distributed representations of code. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–29, 2019.

[3] J. R. Anderson and B. J. Reiser. The lisp tutor. *Byte*, 10(4):159–175, 1985.

[4] A. Badrinath, F. Wang, and Z. Pardos. pybkt: An accessible python library of bayesian knowledge

tracing models. In *In: Proceedings of the 14th International Conference on Educational Data Mining (EDM 2021)*. ERIC, 2021.

[5] R. S. Baker. Gaming the system: A retrospective look. *Philippine Computing Journal*, 6(2):9–13, 2011.

[6] R. S. Baker, A. T. Corbett, and V. Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *International conference on Intelligent Tutoring Systems*, pages 406–415. Springer, 2008.

[7] T. Barnes. The q-matrix method: Mining student response data for knowledge. In *American Association for Artificial Intelligence 2005 Educational Data Mining Workshop*, pages 1–8. AAAI Press, Pittsburgh, PA, USA, 2005.

[8] Y. Bergner, S. Droschler, G. Kortemeyer, S. Rayyan, D. Seaton, and D. E. Pritchard. Model-based collaborative filtering analysis of student response data: Machine-learning item response theory. *International Educational Data Mining Society*, 2012.

[9] H. Cen, K. Koedinger, and B. Junker. Learning factors analysis–a general method for cognitive model evaluation and improvement. In *International conference on intelligent tutoring systems*, pages 164–175. Springer, 2006.

[10] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[11] R. E. Clark, D. F. Feldon, J. J. van Merriënboer, K. A. Yates, and S. Early. Cognitive task analysis. In *Handbook of research on educational communications and technology*, pages 577–593. Routledge, 2008.

[12] C. Conati and K. VanLehn. A student model to assess self-explanation while learning from examples. In *UM99 User Modeling*, pages 303–305. Springer, 1999.

[13] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-adapted Interaction*, 4(4):253–278, 1994.

[14] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, 2005.

[15] M. C. Desmarais and R. Naceur. A matrix factorization method for mapping items to skills and for enhancing expert-based q-matrices. In H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, editors, *Artificial Intelligence in Education*, pages 441–450, Berlin, Germany, 2013. Springer.

[16] M. C. Desmarais and R. Naceur. A matrix factorization method for mapping items to skills and for enhancing expert-based q-matrices. In *International Conference on Artificial Intelligence in Education*, pages 441–450. Springer, 2013.

[17] M. Du, N. Liu, and X. Hu. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77, 2019.

[18] S. H. Edwards and K. P. Murali. Codeworkout: short programming exercises with built-in data collection. In *Proceedings of the 2017 ACM conference on innovation and technology in computer science education*, pages 188–193, 2017.

[19] B. Fein, I. Graßl, F. Beck, and G. Fraser. An evaluation of code2vec embeddings for scratch. In *In Proceedings of the 15th International Conference on Educational Data Mining (EDM) 2022*, 2022.

[20] S. Guerriero. Teachers' pedagogical knowledge and the teaching profession. *Teaching and Teacher Education*, 2(1):7, 2014.

[21] L. Gusukuma, A. C. Bart, D. Kafura, J. Ernst, and K. Cennamo. Instructional design+ knowledge components: A systematic method for refining instruction. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, pages 338–343, 2018.

[22] M. Hoq, P. Brusilovsky, and B. Akram. Analysis of an explainable student performance prediction model in an introductory programming course. In *Proceedings of the 16th International Conference on Educational Data Mining (EDM) 2023*, 2023.

[23] K. R. Koedinger, A. T. Corbett, and C. Perfetti. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive science*, 36(5):757–798, 2012.

[24] K. R. Koedinger, E. A. McLaughlin, and J. C. Stamper. Automated student model improvement. *International Educational Data Mining Society*, 2012.

[25] A. S. Lan, A. E. Waters, C. Studer, and R. G. Baraniuk. Sparse factor analysis for learning and content analytics. *Journal of Machine Learning Research (JMLR)*, 15(57):1959–2008, 2014.

[26] J. Leinonen. Open ide action log dataset from a cs1 mooc. In *Proceedings of the 6th Educational Data Mining in Computer Science Education (CSEDM) Workshop*, 2022.

[27] Y. Mao, F. Khoshnevisan, T. Price, T. Barnes, and M. Chi. Cross-lingual adversarial domain adaptation for novice programming. 2022.

[28] Y. Mao, S. Marwan, T. W. Price, T. Barnes, and M. Chi. What time is it? student modeling needs to know. In *In proceedings of the 13th International Conference on Educational Data Mining*, 2020.

[29] Y. Mao, Y. Shi, S. Marwan, T. W. Price, T. Barnes, and M. Chi. Knowing" when" and" where": Temporal-astnn for student learning progression in novice programming tasks. In *In: Proceedings of the 14th International Conference on Educational Data Mining (EDM 2021)*, 2021.

[30] Y. Mao, R. Zhi, F. Khoshnevisan, T. W. Price, T. Barnes, and M. Chi. One minute is enough: Early prediction of student success and event-level difficulty during a novice programming task. In *Proceedings of the 12th International Conference on Educational Data Mining (EDM 2019)*, 2019.

[31] S. Marwan, Y. Shi, I. Menezes, M. Chi, T. Barnes, and T. W. Price. Just a few expert constraints can help: Humanizing data-driven subgoal detection for novice programming. *International Educational Data Mining Society*, 2021.

[32] M. J. Nathan, K. R. Koedinger, M. W. Alibali, et al. Expert blind spot: When content knowledge eclipses pedagogical content knowledge. In *Proceedings of the*

third international conference on cognitive science, volume 644648, 2001.

[33] B. Paaßen, M. Dywel, M. Fleckenstein, and N. Pinkwart. Sparse factor autoencoders for item response theory. In *Proceedings of the 15th International Conference on Educational Data Mining*, pages 17—-26, Durham, UK, 2022. EDM.

[34] Z. A. Pardos, A. Dadu, et al. dafm: Fusing psychometric and connectionist modeling for q-matrix refinement. *Journal of Educational Data Mining*, 10(2):1–27, 2018.

[35] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 255–266. Springer, 2010.

[36] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[37] P. Pavlik Jr, H. Cen, and K. Koedinger. Performance factors analysis–a new alternative to knowledge tracing. In *Frontiers in Artificial Intelligence and Applications*, volume 200, pages 531–538, Amsterdam, Netherlands, 01 2009. IOS Press.

[38] P. I. Pavlik Jr, H. Cen, and K. R. Koedinger. Learning factors transfer analysis: Using learning curve analysis to automatically generate domain models. *Online Submission*, 2009.

[39] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. *Advances in Neural Information Processing Systems*, 28, 2015.

[40] T. Price, R. Zhi, and T. Barnes. Evaluation of a data-driven feedback algorithm for open-ended programming. *International Educational Data Mining Society*, 2017.

[41] T. W. Price, Y. Dong, and D. Lipovac. isnap: towards intelligent tutoring in novice programming environments. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, pages 483–488, 2017.

[42] T. W. Price, D. Hovemeyer, K. Rivers, G. Gao, A. C. Bart, A. M. Kazerouni, B. A. Becker, A. Petersen, L. Gusukuma, S. H. Edwards, et al. Progsnap2: A flexible format for programming process data. In *ITiCSE'20*, pages 356–362, 2020.

[43] T. W. Price, R. Zhi, and T. Barnes. Hint generation under uncertainty: The effect of hint quality on help-seeking behavior. In *International conference on artificial intelligence in education*, pages 311–322. Springer, 2017.

[44] K. Rivers, E. Harpstead, and K. Koedinger. Learning curve analysis for programming: Which concepts do students struggle with? In *Proceedings of the 2016 ACM Conference on International Computing Education Research*, pages 143–151, 2016.

[45] R. J. Salden, V. A. Aleven, A. Renkl, and R. Schwonke. Worked examples and tutored problem solving: redundant or synergistic forms of support? *Topics in Cognitive Science*, 1(1):203–213, 2009.

[46] S. Sarsa, P. Denny, A. Hellas, and J. Leinonen. Automatic generation of programming exercises and code explanations using large language models. In *Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 1*, pages 27–43, 2022.

[47] R. Schmucker, J. Wang, S. Hu, T. Mitchell, et al. Assessing the knowledge state of online students-new data, new approaches, improved accuracy. *Journal of Educational Data Mining*, 14(1):1–45, 2022.

[48] D. Selent, T. Patikorn, and N. Heffernan. Assistments dataset from multiple randomized controlled experiments. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pages 181–184, 2016.

[49] Y. Shi, M. Chi, T. Barnes, and T. Price. Code-dkt: A code-based knowledge tracing model for programming tasks. In *In Proceedings of the 15th International Conference on Educational Data Mining (EDM) 2022*, 2022.

[50] Y. Shi, Y. Mao, T. Barnes, M. Chi, and T. W. Price. More with less: Exploring how to use deep learning effectively through semi-supervised learning for automatic bug detection in student code. In *EDM'21*, 2021.

[51] Y. Shi and T. Price. An overview of code2vec in student modeling for programming education. *MMTC Communications-Frontiers*, 2022.

[52] Y. Shi, K. Shah, W. Wang, S. Marwan, P. Penmetsa, and T. Price. Toward semi-automatic misconception discovery using code embeddings. In *LAK21: 11th International Learning Analytics and Knowledge Conference*, pages 606–612, 2021.

[53] G. S. Snoddy. Learning and stability: a psychophysiological analysis of a case of motor learning with clinical applications. *Journal of Applied Psychology*, 10(1):1, 1926.

[54] P. Sztajn, J. Confrey, P. H. Wilson, and C. Edgington. Learning trajectory based instruction: Toward a theory of teaching. *Educational researcher*, 41(5):147–156, 2012.

[55] S. Yang, X. Liu, H. Su, M. Zhu, and X. Lu. Deep knowledge tracing with learning curves. In *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 282–291. IEEE, 2022.

[56] M. Yudelson, R. Hosseini, A. Vihavainen, and P. Brusilovsky. Investigating automated student modeling in a java mooc. In *In Proceedings of the 7th International Conference on Educational Data Mining (EDM) 2014*, 2014.

[57] J. Zhang, X. Wang, H. Zhang, H. Sun, K. Wang, and X. Liu. A novel neural source code representation based on abstract syntax tree. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 783–794. IEEE, 2019.

[58] Y. Zhang, J. D. Pinto, A. X. Fan, L. Paquette, et al. Using problem similarity-and orderbased weighting to model learner performance in introductory computer science problems. *Journal of Educational Data Mining*, 15(1):63–99, 2023.

Learning Problem Decomposition-Recomposition with Data-driven Chunky Parsons Problems within an Intelligent Logic Tutor[*]

Preya Shabrina[†]
North Carolina State University
Raleigh, NC, USA
pshabri@ncsu.edu

Behrooz Mostafavi
North Carolina State University
Raleigh, NC, USA
bzmostaf@ncsu.edu

Sutapa Dey Tithi
North Carolina State University
Raleigh, NC, USA
stithi@ncsu.edu

Min Chi
North Carolina State University
Raleigh, NC, USA
mchi@ncsu.edu

Tiffany Barnes
North Carolina State University
Raleigh, NC, USA
tmbarnes@ncsu.edu

ABSTRACT

Problem decomposition into sub-problems or subgoals and recomposition of the solutions to the subgoals into one complete solution is a common strategy to reduce difficulties in structured problem solving. In this study, we use a data-driven graph-mining-based method to decompose historical student solutions of logic-proof problems into *Chunks*. We design a new problem type where we present these chunks in a *Parsons Problem* fashion and asked students to reconstruct the complete solution from the chunks. We incorporated these problems within an intelligent logic tutor and called them *Chunky Parsons Problems* (CPP). These problems demonstrate the process of problem decomposition to students and require them to pay attention to the decomposed solution while they reconstruct the complete solution. The aim of introducing CPP was to improve students' problem-solving skills and performance by improving their decomposition-recomposition skills without significantly increasing training difficulty. Our analysis showed that CPPs could be as easy as Worked Examples (WE). And, students who received CPP with simple explanations attached to the chunks had marginally higher scores than those who received CPPs without explanation or did not receive them. Also, the normalized learning gain of these students shifted more towards the positive side than other students. Finally, as we looked into their proof-construction traces in posttest problems, we observed them to form identifiable chunks aligned with those found in historical solutions with higher efficiency.

Keywords

Parsons Problem, Intelligent Tutors, Data-driven Subgoal, Problem Decomposition

1. INTRODUCTION

Computational thinking, a set of skills and practices for complex problem solving, provides a foundation for learning 21st-century skills, particularly computer science (CS). Educational researchers and teaching professionals acknowledge problem decomposition-recomposition skill as a key component of computational thinking and complex problem solving [11, 23, 13, 22]. Efficient problem solving using the problem decomposition skill or strategy involves several steps: 1) identifying sub-problems (i.e. subgoals) to reduce the difficulty associated with the problem, 2) constructing a solution for each of those sub-problems, and 3) recomposing the sub-problem solutions to form the larger solution [5]. Research showed that experts carry out problem decomposition and recomposition (PDR) steps more than novices [41]. However, several studies also showed that novices often attempt to decompose problems [26, 40]. But while they may demonstrate correct decomposition in easier problems, novices fail to decompose sophisticated problems [26].

Despite problem decomposition-recomposition (PDR) being vital to complex problem-solving, it is rarely mentioned explicitly in instructional materials for computer science (a discipline focused on complex problem solving using computers) [30]. Also, existing research lacks guidance on how to motivate students to adopt this PDR process or how to improve their skills associated with PDR. A few studies analyzed the differences between experts and novices in adopting this PDR process, indicating that experts use PDR more than novices [27, 20]. And, a few studies aimed at introducing this PDR skill to students, mostly during programming problem solving, using varying methods [for example, us-

[†] First Author (led and carried out most of the work).

P. Shabrina, B. Mostafavi, S. D. Tithi, M. Chi, and T. Barnes. Learning problem decomposition-recomposition with data-driven chunky parsons problems within an intelligent logic tutor. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th Interna-tional Conference on Educational Data Mining*, pages 40–54, Ben-galuru, India, July 2023. International Educational Data Mining So-ciety.

https://doi.org/10.5281/zenodo.8115716

ing pattern-oriented instruction [32], programming problem decomposition exercise [42], guided inquiry-based instruction [36], etc.]. However, PDR remains under-explored in the instruction of other structured problem-solving domains.

In this study, we design, implement, and evaluate a problem-based training intervention, named *Chunky Parsons Problem* (CPP), that introduces to students the concept of problem decomposition-recomposition (PDR) while engaging them in these processes during problem solving within an intelligent logic tutor, DT (Deep Thought). To generate CPP, we decomposed students' historical solutions to each logic-proof construction problem stored in DT's problem bank into subproofs (referred to as ***chunks***) using a data-driven method. These chunks are presented in a *Parsons Problem* fashion. In a traditional Parsons Problem, all steps contributing to the complete solution of a problem are presented in a jumbled order. On the contrary, within CPP, the solution to a logic-proof problem is presented as jumbled-up chunks (groups of connected statements) instead of individual statements. By design, CPP is a partially worked example where all the required statements are shown in chunks. However, the missing connections among the chunks give the students an opportunity to recompose the solution while having them pay attention to the decomposition to understand the composition of each chunk, how each chunk contributes to other chunks, and the overall solution. Thus, CPPs can be thought of as problems that are partially worked examples and partially problem-solving (PS) problems.

We deployed DT with CPP implemented within its training session in an undergraduate classroom of CS majors and conducted a controlled experiment. In the controlled experiment, we implemented three training conditions: 1) Control(C): received only worked example (WE) and problem-solving (PS) logic-proof construction problems, 2) Treatment 1(T_1): received CPP (**without** explicit explanation of the chunks) along with PS/WE, and 3) group who receive CPP (**with** explicit explanation attached to the chunks) along with PS/WE. Since prior research showed that explicit instruction on what to learn or take away from an intervention may help to improve students' decomposition ability [36], we introduced the last training condition to identify the more effective representation (between with/without explanation) of CPP. Finally, we evaluated the efficacy of CPP by answering the following research questions:

- ***RQ1:*** How do *Chunky Parsons Problems* impact students' performance and learning?
- ***RQ2:*** What are the difficulties associated with solving a *Chunky Parsons Problem*?
- ***RQ3:*** How do *Chunky Parsons Problems* impact students' Chunking (problem decomposition-recomposition) behavior and skills while solving a new problem?

2. BACKGROUND AND MOTIVATION

Existing research has identified problem decomposition-recomposition (PDR) as difficult for novices as problems get more complex [26]. However, we found only a few studies investigating methods to improve this skill. For example, Pearce et al. [36] explored explicit instruction (openly instructing students to learn problem decomposition and describing how to go about that learning) to improve students' problem decomposition skills and concluded that explicit instruction can lead to significant gains in mastering this skill. Muller et al. [32] found that pattern-oriented instruction can have a positive impact on problem decomposition skills. We found some studies where researchers in the domain of mathematics and programming, using problem-based methods, aimed at improving students' subgoal learning which is equivalent to the skill of identifying sub-tasks required to solve a problem (i.e. problem decomposition). The most common method explored by researchers in this regard is subgoal-labeled worked examples or instructional materials [29, 8, 7]. Studies showed that worked examples with abstract labels that give away structural information help improve students' problem-solving skills measured by test scores. However. these studies do not evaluate or measure students' problem decomposition or subgoaling skills after training. Also, we did not find any established guidance on how problem-based interventions can be generated automatically and how they should be designed to be used within tutors to improve students' problem decomposition-recomposition (or chunking) skills.

From our literature review, we concluded that problem-based interventions specifically designed for tutors to improve students' chunking skills are under-explored. Thus, in this paper, we set our aim to design and implement CPPs to be used within DT to improve students' problem-solving and chunking skills. While extracting chunks to present within CPP and designing its representation within DT, we considered three goals: 1) Automating the solution-decomposition process to extract chunks so that expert effort is not required; 2) Designing the problem to demonstrate chunking and engaging students in the process to improve their skills, and 3) keeping the difficulty-level low so that students can persist and learn. To set the difficulty level of our problem, we explored problem types that are of low difficulty as established by literature: Worked Examples and Parsons Problems.

Worked Examples: Worked examples (WE) reduce learners' intrinsic load (i.e. working memory load which is caused by the complexity of the problem) and help them to learn better [35]. This improvement in learning due to worked examples is referred to as the Worked Example Effect [44] in literature. However, several studies argued the applicability of worked examples in certain situations. For example, worked examples may not be useful for students with high prior knowledge [34], when problems are structured [34], or if the problem is strategic but involves only a few interactive elements [10]. In such cases, problem-solving (PS) supports the learning process better [10]. Also, for goal or product-directed problems, a worked example only shows the construction of the solution and does not help students to grow an understanding of the rationale behind the selection of certain steps [49]. In this scenario, students fail to acquire a schema of the problem-solving approach which leads to the failure to transfer problem-solving skills. Renkl et al. [39] suggested that worked examples help students to learn better only when the examples give away structural information of the solution and isolate meaningful building blocks.

Parsons Problems: Parsons problems ask students to construct a solution from a given set of jumbled solution steps [14]. Poulsen et al. showed the application of the Parsons problem in a mathematical proof construction tool, Proof Blocks [38].

They found that Parsons Problems within Proof Blocks significantly reduced the difficulty associated with proof construction. Parsons problem is heavily explored in programming education. Studies found that Parsons problems can improve students' code writing capability [48, 24, 14, 17] or can help in completing programming tasks efficiently without impacting performance on subsequent programming tasks [51]. Studies also showed that attached explanations [18] and subgoal labels [31] can help students solve Parsons problem and improve the learning process.

From the overview of the impact of WEs and Parsons Problem, we designed CPPs as partially WE and partially PS, which represent meaningful building blocks of a proof (i.e. Chunks) in a Parsons Problem fashion. Additionally, we explored attaching explanations to the chunks to further reduce difficulties and support students' learning process.

Data-driven Solution Decomposition Techniques: Data-driven solution decomposition refers to the process of automatically decomposing a problem or its solution into subgoals or subsolutions based on the properties found in historical solutions. These historical solutions often come from tutors or learning platforms that collect students' solution traces and thus, are often redundant. While performing data-driven decomposition, researchers mainly focused on identifying independent or dependent components of a solution. To do so, they often presented student solutions as graphs depicting how they moved from state to state to reach the final solution [37, 50, 45, 33, 4]. Prior research showed the application of clustering [37] or connected component detection [50, 45, 16] to extract independent sub-solutions or chunks in these graphical models. On the other hand, some researchers [12, 19] proposed constraint-based decomposition techniques for linear problem solutions such that decomposed sub-solutions can be replaced with alternate solutions without causing any problem. To decompose computer programs, researchers have used methods where they looked at the usage of different program components (for ex., variables) to identify independent parts of the program [46, 47]. In this paper, we demonstrate a solution decomposition method that extracts chunks by applying rules/constraints on graphical representations of historical student solutions similar to Eagle et al.'s work [16].

Evaluation of Students' Chunking Skills We observed that in prior research, researchers have only used test scores to evaluate methods that were set to teach students chunking/PDR. Only a few recent studies explored methods to measure students' chunking skills from data. Kwon and Cheon [26] mapped predefined sub-tasks and program segments in Scratch programs to observe how students decompose and develop programs. In a recent study, Charitsis et al. [9] used NLP to identify key components and students' approaches to develop programs and then relate those to performance metrics using linear regression to quantify their decomposition skills. Kinnebrew et al. [25] also mined frequent patterns in students' action sequences and relate that to performance to explain their learning behavior. Overall, to evaluate decomposition skills, these studies each sought a baseline to compare students' solutions against and explained performance using solution characteristics. In this paper, to measure students' chunking/PDR skills after being trained with CPPs, we analyzed students' step sequences during proof construction to identify potential chunking/PDR instances and tried to explain their performance through the chunking/PDR characteristics.

3. METHOD

In this study, we explored Chunky Parsons Problems (with or without explicit explanations attached to them) to improve students' problem-solving skills (with an emphasis on *Chunking/PDR* skills) and learning gain in the context of logic-proof problems. We derived CPP using a data-driven method and incorporated them into the training session within DT [28], an intelligent logic tutor. In the subsequent sections, we first provide a brief introduction to DT. Then, we discuss how we derived CPP from data, designed explanations explaining the chunks, and presented them within DT. Finally, we present the design of our experimental training conditions and data collection method to facilitate analyses to answer our research questions

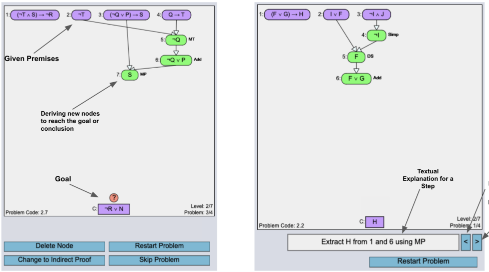
3.1 Deep Thought (DT), the Intelligent Logic Tutor

DT is an intelligent logic tutor that teaches students logic-proof construction. Each logic-proof problem within DT contains a set of given premises and a conclusion presented as visual nodes [Figure 1a]. To solve a problem, new propositions (or nodes) are needed to be derived by applying valid logic rules on the given premises and subsequently on derived premises to reach the conclusion. Usually, each problem in DT is either of type Worked Example (WE) or problem-solving (PS). WEs are solved by the tutor step-by-step as the students click on a next step (>) button [Figure 1b]. On the other hand, PSs are required to be solved by the students where they have to derive all the steps of a proof [Figure 1a]. Here, a step refers to the process of deriving a single node or proposition.

DT is organized into 7 levels. In the first level, the tutor starts by showing two sample logic-proof problems (one WE and one PS) to help students understand how to use different features of the tutor. Then, the students solve two pretest PS problems. After the pretest level (i.e. level 1), the students go through 5 training levels with 4 problems in each level. Each of the first three problems in the training levels is either a WE or PS. For these training-level PS problems, on-demand step-level hints are available. The last problem in each training level is always of type PS and is called the training-level test problem. After the 5 training levels, students enter into a posttest level containing 6 PS problems. During the pretest, training-level test, and posttest problems, the tutor does not offer any hints or help and the students have to solve them independently. For each of these problems, students receive a score between 0 and 100 (efficient proof construction [less time, fewer step counts, and incorrect rule applications] receives higher scores) [3, 1]. The pretest scores represent students' mastery level before training. On the other hand, the training-level posttest and posttest scores track how much students learned after each level of training and after all 5 training levels. More Details on DT interface and features can be found in Appendix A.

3.2 Deriving Chunky Parsons Problem (CPP) using a data-driven Graph-Mining Approach

Data for Deriving CPP: DT has been being deployed in an undergraduate logic course offered at a public research uni-



Extract H from 1 and 6 using MP

a) Problem Solving or PS

b) Worked Example or WE

Figure 1: (a) PS and (b) WE Interface in DT

versity in the Fall and Spring semesters since 2012. To derive CPP representation for logic proofs, we used the most recent log data collected by DT in the Fall and Spring of the years 2018-2021. These log data detail students' historical step-by-step proof-construction attempts for each problem they solved within DT. Using these data, we generated high-level graphical representations (Interaction Networks [16] and Approach Maps [15]) of students' solution approaches for these problems to derive CPP from them. For each problem, data of approximately 170-200 students containing altogether 2000-5500 solution steps were used. To select the students, we performed equal random sampling from the semesters mentioned before so that the data is representative of different student groups who took the course over the years. The reason for not using all data is to mainly reduce the computational complexity of the adopted graph mining approach. Also, the data used is assumed sufficient enough to capture common student approaches to solve each logic-proof problem in the problem bank of DT [43].

Interaction Network and Approach Map: For each of the problems in the DT problem bank, we generated a graphical representation of how students moved from one state to another during the construction of a proof for the problem. Here, a state refers to all nodes (or propositions) a student had at a particular moment during their proof construction attempt. Students move from state to state by deriving or deleting nodes, i.e. via a step. To limit the number of states in the graph, the propositions at a particular state are lexicographically ordered, which means that the order of derivation of the nodes is not considered in the graphical representation. This graphical representation of students' proof-construction attempts for a problem is called an interaction network since it represents the interaction among the states [16]. Since interaction networks are often very large and visually uninterpretable, we applied Girvan-Newman community clustering [21] on the interaction networks to identify regions or clusters of closely connected states. Each cluster contains a set of states containing effective propositions that contributed to the final proof submitted by the students and also unnecessary propositions (i.e. propositions that did not contribute to the final proof) that they derived along the way. We represented each cluster with one single graphical node containing only the effective propositions. Thus, we obtained a graph where the start state containing the given premises is connected to the conclusion through clusters of effective propositions. Each path from start to conclusion represents one student approach (or solution) to the logic-proof problem [sample approach maps are visualized in Figure 2]. Thus, this representation is called an approach map [15]. Later, we used a rule-based approach to extract chunks from the approach maps.

Extracting Chunks from Approach Maps: As discussed in Section 2, researchers [12, 19] have decomposed problem solutions using constraints such that the decomposed sub-solutions can be replaced with alternate solutions without causing any problem. Based on this idea, we defined two rules to extract pivot[1] or subgoal propositions that are present in multiple approaches and/or have multiple replaceable derivations within an approach map (for example, $\neg K \vee N$ in Figure 2a has two possible derivations from the start state.). The rules to identify such pivots are:

Rule 1: First proposition derived within a cluster where multiple clusters merge is a pivot [$\neg K \vee N$ in Figure 2a].

Rule 2: Last proposition derived within a cluster that generates a fork is a pivot [$\neg K \vee N$ and $\neg(K \wedge \neg N)$ in Figure 2a].

Recall that an approach is a path from start to goal in an approach map. And, being present in multiple paths or approaches means that a proposition is possibly vital to the proof and a subgoal in student approaches. Finally, we defined a third rule to identify pivots in approaches that do not have a common proposition with other approaches, i.e. they are simply a linear chain of clusters of propositions[Figure 2b]. The third rule is described below:

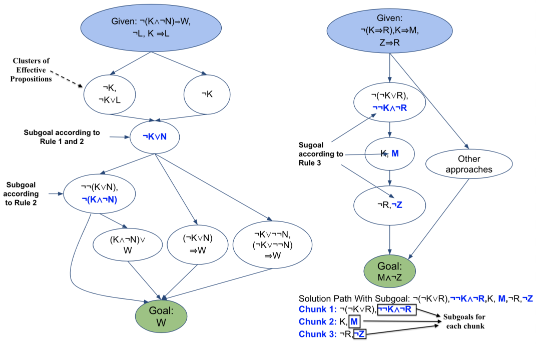
Rule 3: In a chain of clusters, the last derived node in each cluster is a pivot [M or $\neg Z$ in Figure 2b]. Note that in this rule, we simply exploit the clusters identified by Girvan-Newman algorithm to dismantle a complete solution into sub-solutions or subgoals.

Finally, Using the three rules, we extracted the subgoals within the most common student-solution approach for each DT logic-proof problem while traversing its approach map from top to bottom. We validated our pivot/subgoal- extraction process by comparing our rule-based subgoals from approach maps against expert-identified[2] subgoals for 15 problems. And, our method was successful in identifying all expert subgoals for those problems. After validation, we used the subgoals to decompose the solution to derive *Chunks* from them. An example of deriving chunks can be found in Figure 2b. In the example, pivots/subgoals are colored blue, and using the subgoals three chunks are extracted from the complete solution. Note here that each chunk is associated with a subgoal.

Explanations for Chunks: To accompany each of the chunks, we generated automated explanations using a script that explains the composition and purpose of the chunks. Before writing the script, a format for the chunk explanations was decided through discussion with an expert. Each explanation is written in natural language and tells what a chunk derives (i.e. the associated subgoal), how the subgoal is derived within the chunk, and why it is derived [Figure 3b]. The why part simply tells that each subgoal is necessary for the derivation of another subgoal or the final goal. Note that we paid close attention while crafting the explanation format so that it does not give away any information about the final solution beyond the visual representation of the chunks. Overall, the purpose of the explanations is just to

[1] major propositions within a proof that can be used to decompose the proof, also referred to as *Subgoals.*

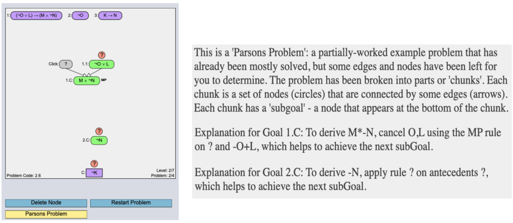
[2] The experts are two academic professionals with 10+ years of experience with logical reasoning



a) Approach Map Example 1

b) Approach Map Example 2 and Chunking

Figure 2: Demonstration of a) Rule 1 and 2; and b) Rule 3 and Chunk Extraction



This is a 'Parsons Problem': a partially-worked example problem that has already been mostly solved, but some edges and nodes have been left for you to determine. The problem has been broken into parts or 'chunks'. Each chunk is a set of nodes (circles) that are connected by some edges (arrows). Each chunk has a 'subgoal' - a node that appears at the bottom of the chunk.

Explanation for Goal 1.C: To derive M*-N, cancel O,L using the MP rule on ? and -O+L, which helps to achieve the next subGoal.

Explanation for Goal 2.C: To derive -N, apply rule ? on antecedents ?, which helps to achieve the next subGoal.

a) Chunky Parsons Problem Interface

(b) Chunky Parsons Problem Explanation

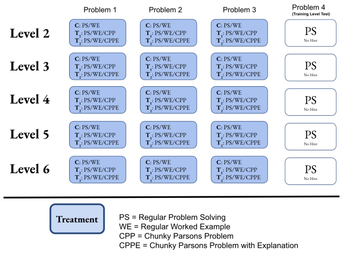
Figure 3: a) Parsons Problem Interface in DT; b) Explanation Given to Specific Student Groups for Chunks Presented in a Parsons Problem.

highlight what the chunks represent (i.e. they are building blocks of a complete solution each deriving a subgoal).

3.2.1 Chunky Parsons Problem Interface

The Chunky Parsons Problem representation is shown in Figure 3a. In the presentation, the given premises and conclusion are presented as usual. And the chunks are presented as groups of connected propositions (or nodes) in a Parsons Problem fashion. The problem shown in Figure 3a has two chunks. All nodes within the chunks are connected to each other. However, the givens, the chunks, and the conclusion need to be connected by students to complete the proof. Each node within a chunk can be either justified (both antecedents present), partially justified (one of the antecedents missing as for $M \wedge \neg N$), or unjustified (all of the antecedents missing as for $\neg O \vee L$ or $\neg N$). For justified and partially justified nodes, the associated logic rule is also shown. For example, in Figure 3a, $M \wedge \neg N$ is labeled by MP, i.e. Modus Ponens is required for its derivation from the antecedents. In addition to the visual components, textual instructions containing chunk explanations [Figure 3b] were also provided to students who were assigned to a specific training condition (more details about training conditions in the next subsection). Note that the chunk or subgoal IDs (for example, 1.C, 2.C, etc. in the figure) are used to associate an explanation to a chunk and the IDs do not confirm the order of how the chunks should be connected to each other.

3.3 Experiment Design



	Problem 1	Problem 2	Problem 3	Problem 4 (Training Level Test)
Level 2	**C**: PS/WE $\mathbf{T_1}$: PS/WE/CPP $\mathbf{T_2}$: PS/WE/CPPE	**C**: PS/WE $\mathbf{T_1}$: PS/WE/CPP $\mathbf{T_2}$: PS/WE/CPPE	**C**: PS/WE $\mathbf{T_1}$: PS/WE/CPP $\mathbf{T_2}$: PS/WE/CPPE	PS No Hint
Level 3	**C**: PS/WE $\mathbf{T_1}$: PS/WE/CPP $\mathbf{T_2}$: PS/WE/CPPE	**C**: PS/WE $\mathbf{T_1}$: PS/WE/CPP $\mathbf{T_2}$: PS/WE/CPPE	**C**: PS/WE $\mathbf{T_1}$: PS/WE/CPP $\mathbf{T_2}$: PS/WE/CPPE	PS No Hint
Level 4	**C**: PS/WE $\mathbf{T_1}$: PS/WE/CPP $\mathbf{T_2}$: PS/WE/CPPE	**C**: PS/WE $\mathbf{T_1}$: PS/WE/CPP $\mathbf{T_2}$: PS/WE/CPPE	**C**: PS/WE $\mathbf{T_1}$: PS/WE/CPP $\mathbf{T_2}$: PS/WE/CPPE	PS No Hint
Level 5	**C**: PS/WE $\mathbf{T_1}$: PS/WE/CPP $\mathbf{T_2}$: PS/WE/CPPE	**C**: PS/WE $\mathbf{T_1}$: PS/WE/CPP $\mathbf{T_2}$: PS/WE/CPPE	**C**: PS/WE $\mathbf{T_1}$: PS/WE/CPP $\mathbf{T_2}$: PS/WE/CPPE	PS No Hint
Level 6	**C**: PS/WE $\mathbf{T_1}$: PS/WE/CPP $\mathbf{T_2}$: PS/WE/CPPE	**C**: PS/WE $\mathbf{T_1}$: PS/WE/CPP $\mathbf{T_2}$: PS/WE/CPPE	**C**: PS/WE $\mathbf{T_1}$: PS/WE/CPP $\mathbf{T_2}$: PS/WE/CPPE	PS No Hint

PS = Regular Problem Solving
WE = Regular Worked Example
CPP = Chunky Parsons Problem
CPPE = Chunky Parsons Problem with Explanation

Figure 4: Problem Organization in the Training Levels for the Three Training Conditions. Note: '/' indicates a random selection. For example, 'PS/WE/CPP' indicates that the problem will be randomly presented as either a PS, or a WE, or a CPP.

Using existing problem types within DT (PS and WE) and the new problem types (CPP), we designed three training conditions. The three training conditions are described below:

Control (C): Students assigned to the Control (C) condition received only PS or WE (selected randomly) during training.

Treatment 1 (T_1): Students assigned to this condition, may receive CPP without explanation in addition to PS/WE (selected randomly) during training.

Treatment 2 (T_2): Students assigned to this condition may receive CPP with an explanation (i.e. CPPE) in addition to PS/WE (selected randomly) during training.

Problem organization for each condition in the 5 DT training levels is demonstrated in Figure 4. Note that the Control (C) condition gives us a baseline for comparison between students who received CPP or CPPE [i.e. T_1/T_2 students]) and those who did not receive CPP at all (i.e. C students). On the other hand, a comparison between T_1 and T_2 helps to understand the impact of the explicit explanation attached to each chunk in a CPP.

System Deployment and Data Collection: We deployed DT with the three training conditions in an undergraduate logic course offered at a public research university in the Spring of 2022. Each participating student in that course was assigned to one of the three training conditions after they completed the pretest problems. Our training condition assignment algorithm ensures that the pretest scores of students in each of the training groups have a similar distribution. Finally, we had 50 students assigned to C, 50 students assigned to T_1, and 45 students assigned to T_2 who completed all 7 levels (pretest, all training levels, and the posttest level) of the tutor. We collected their pretest, training-level test, and posttest scores to compare performance/learning across the training groups. Additionally, we collected their solution traces to analyze differences in their proof construction approaches. Note that access to these data is restricted to IRB-authorized researchers. To answer our research questions, we carried out statistical and data-driven graph-mining-based analyses on the collected data that we report in the subsequent sections.

4. RESULTS

4.1 RQ1: Students' Performance and Learning Gain

To understand the impact of each of our training conditions on students' performance and learning, we analyzed students' test score-based performance and normalized learning gain (NLG) after training. For these analyses, we focused on the training-level test problems (2.4-6.4) and the posttest problems (7.1-7.6) that students solved independently without any tutor help. We adopted a combination of regression and statistical analysis (Kruskal-Walis test with posthoc pairwise Mann-Whitney test with Bonferroni corrected $\alpha = 0.016^3$) to compare the performance and learning gain across the three training conditions. These tests do not make an assumption about the data being perfectly normal. Since most of our collected data were skewed, these tests were considered suitable in this case. Note that there were no significant differences found in performance across the three groups in the pretest problems.

4.1.1 *Test Score-based Performance*

To identify the association between the training conditions and performance, we performed two mixed-effect regression analyses: one for the training-level test problems and one for the posttest problems. In each of these two analyses, problem IDs were defined as the random-effect variable (to eliminate the impact of differences across problems), training conditions were defined as the fixed-effect variable, and problem score was the dependent variable. The analysis for the training-level test problems [avg. training-level test scores(C, T_1, T_2) = 65.1, 61.7, and 65.5] gave a p-value of 0.8 [$p < 0.05$ indicates significance] indicating that there was no significant association between training-level test performance and the training conditions. However, the analysis for the posttest problems [avg. posttest scores(C, T_1, T_2) = 69.3, 68.2, and 73.5] gave a p-value of 0.06 demonstrating a marginally significant association between the training conditions and posttest performance. Also, the average posttest scores showed that T_2 (who received CPPE) marginally outperformed the other two groups after 5 levels of training. To further investigate each training group's posttest performance, we statistically compared scores across the three training groups in each of the independent posttest problem-solving instances (7.1-7.6). The trend in scores for these problems across the three training groups is shown in Figure 5. While analyzing the scores in the posttest problems, we observed that T_2 had significantly higher scores than T_1 and C in problems 7.1-7.3 [for 7.1, $P_{MW}(T_2 > T_1)^4 = 0.003$ and $P_{MW}(T_2 > C) = 0.01$, for 7.2, $P_{MW}(T_2 > T_1) = 0.02$ and $P_{MW}(T_2 > C) = 0.01$, and for 7.3, $P_{MW}(T_2 > T_1)$ = 0.03 (marginal) and $P_{MW}(T_2 > C) = 0.012$] and higher average scores in problems 7.4-7.6. Note that posttest problems in DT are organized in increasing order of difficulty. Our analyses indicate that even though T_2 could not significantly outperform the other two groups in the harder posttest problems(7.3-7.6), they performed comparatively better. This trend can be observed in the 'Posttest' fragment in Figure 5.

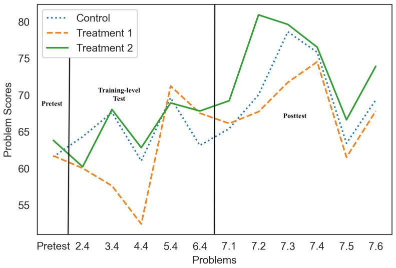


Figure 5: Training-level Test and Posttest Scores across the Three Training Groups

As shown in the figure, although T_2 did not show a significantly higher average than the other two conditions in all problems, starting from problem 6.4, T_2 students always had higher scores (shown by the solid green line) than the other two groups (shown by the dotted blue line and dashed orange line). Overall, from our regression and statistical analysis, we concluded that students' posttest performance was associated with the training conditions, and the T_2 training condition that involved CPPE was more helpful in improving students' performance after training. However, T_2 students showed evidence of improved performance around the end of training and in the posttest rather than showing gradual improvement over the period of training. A consistent pattern that indicates improved performance could not be identified for T_1 students who received CPP without an explanation attached.

4.1.2 *Normalized Learning Gain*

To identify the training condition that was most effective in promoting learning, we analyzed students' normalized learning gain (NLG) across the three training conditions. NLG is defined as the ratio between how much the students learned and the maximum they could have learned between the period of pretest and posttest and is represented by the following equation:

$$NLG = (post - pre)/\sqrt{(100 - pre)} \qquad (1)$$

Note that NLG is normalized between -1 and 1. A negative NLG value represents that the posttest scores are lower than the pretest scores. Negative NLGs could occur if the students did not learn enough from training or if the posttest problems are significantly harder than the pretest problems. NLG for the three groups is shown in Table 1. We compared the NLGs across the three training groups using statistical tests. A Kruskal-Walis test demonstrated significant differences in the NLGs across the three training groups (statistic=5.8, p-val=0.05). As we carried out posthoc pairwise Mann-Whitney U tests with Bonferroni corrected α=0.016, we observed T_2 students had significantly higher NLGs than Control (C) (statistic=1283.0, p-val=0.01) and T_1 (statistic=1235.0, pvalue=0.02) students. As we plotted the distribution of NLGs for the training groups in Figure 6, we observed that the distribution of NLGs for T_2 is centered around positive (+) values, whereas the other two groups had tails on the negative (-) side. Also, as reported in Table 1, 80% of T_2 students had a positive NLG, whereas the percentage for the other two groups are only 70% and 72% respectively. The results of this analysis on NLG indicate that training condition T_2 (combination of CPPE with PS/WE)

[3] In the pairwise tests each datapoint was used in at most three tests: (C, T_1), (C, T_2), and (T_1, T_2). Thus, corrected $\alpha = 0.05/3$

[4] $P_{MW}(T_2 > T_1)$ refers to the p-value obtained from the Mann-Whitney U test for the hypothesis "T_2 had significantly higher values than T_1 for the metric under consideration." $p < 0.016$ indicates significance.

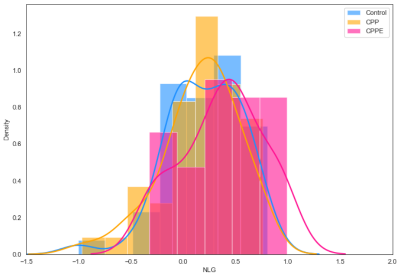


Figure 6: NLG across the Three Training Groups

Table 1: Normalized Learning Gain (NLG) across the Three Training Groups

Group (n)	Pre	Post	NLG	%Student with(+) NLG
C (50)	61.6(19.7)	70.4(14.4)	0.20(0.35)	70%
T1(50)	61.7(18.3)	68.2(15.1)	0.16(0.37)	72%
T2(45)	60.8(18.9)	73.2(14.8)	0.31(0.33)	80%

helped the students to learn better which moved their NLG above 0. Possibly, CPPE helped the students to perform comparatively well even in the harder problems (7.3 to 7.6) that could have caused negative NLG otherwise.

4.2 RQ2: Difficulties Associated with Solving Chunky Parsons Problems

The results from students' performance analysis showed that CPP with explanations attached to chunks (i.e. CPPE) has the potential to improve students' performance and learning gains. However, since it is a new type of problem-based training intervention, we acknowledged the necessity of analyzing its difficulty level in comparison to traditional training interventions like PS or WE. Since tutors like DT are often used by learners in the absence of a human tutor, our aim was to avoid increasing the training difficulty so that the students can persist and learn. Thus, we carried out a comparative analysis between the difficulty level of training CPP/CPPE and PS/WE problems. The difficulties associated with each problem type were measured by the average time that the students needed to solve them (i.e. the problem-solving time). Additionally, to guide future improvements so that the students are better supported during training with CPP/CPPE, we carried out an analysis to identify difficulties that could be associated with specific problem structures where students may need additional help to succeed. In the subsequent sections, we report the findings from the two analyses.

4.2.1 *Comparative Difficulty Level of CPP/CPPE*

To understand the comparative difficulty level of CPP/CPPE, we compared the problem-solving times of CPP/CPPE against the problem-solving times of PS/WE using Mann-Whitney U tests. The plot representing problem-solving times for each of these problem types over the period of training is shown in Figure 7. Notice that in the first two training levels, students' problem-solving time for CPP/CPPE was almost twice the problem-solving time of PS. This higher problem-solving time in the early training levels could be potentially associated with the additional time that the students needed to figure out how different components in the CPP/CPPE interface within DT work. However, as training progressed problem-solving time for CPP became more aligned with that of PS [notice the problem-solving times and comparative p-values at training levels 5 and 6 in Figure 7]. On the other hand, CPPE problem-solving times were marginally or significantly lower than that of PS at levels 5 and 6 respectively. Additionally, CPPE problem-solving times were only marginally higher than that of WEs in these two levels. These statistics indicate that the difficulty level of Chunky Parsons Problems (with/without explanation) lies in between the difficulty levels of PS/WE. However, with explanation, it can be a low-difficulty training task (difficulty level similar to WEs and lower than PS in terms of problem-solving time) that can help improve students' learning gain.

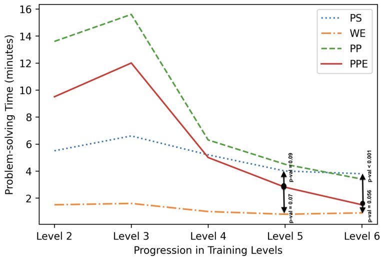
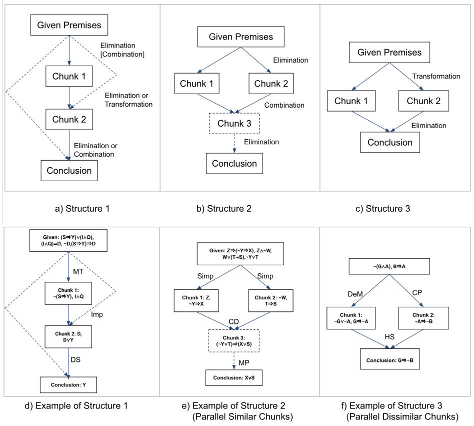


Figure 7: Problem-solving Times for Different Problem Times over the Period of Training.

4.2.2 *Difficulties Associated with Specific Problem Structure*

To identify difficulties associated with specific problem structures, we calculated the average time students spent to complete the proof of each chunk presented in a CPP/CPPE (by connecting all nodes within a chunk to their correct predecessor). We call this chunk-solving time. We identified 10 problems that contained chunks with chunk-solving time above the 75th percentile (> 2.5 minutes) for at least 10% (>= 10 students) of all T_1 (CPP) and T_2 (CPPE) students. To identify the difficulty patterns in these problems, we carried out an exploratory analysis of the structures of these problems and how the students approached to solve the problem. For simplicity, while explaining the problems associated with student difficulties, we present only the abstract structure of the problems [Figure 8a, b, and c]. In the abstract structure, we show how the chunks need to be connected to solve the problem and rule categories instead of the specific rules required to connect the chunks. We grouped the available logic rules in DT into 3 categories: 1) Transformation rules: transform the logic operator in between variables or reorganize the variables in a proposition (Comm, Assoc, DN, De Morgan, Impl, CP, Equiv, Dist), 2) Elimination: remove one or more variables from proposition(s) (MP, MT, DS, Simp, HS), 3) Combination: combines variables from two propositions in one proposition (Add, Conj, CD). For the 10 problems, we identified three abstract problem structures that are shown in Figure 8. Structure 1 was associated with 6 problems. Structures 2 and 3 were asso-



a) Structure 1 b) Structure 2 c) Structure 3

d) Example of Structure 1 e) Example of Structure 2 (Parallel Similar Chunks) f) Example of Structure 3 (Parallel Dissimilar Chunks)

Figure 8: Abstract Structure of Problems where Students Spent Higher Times when Presented as CPP or CPPE. Note: Dashed components are missing in some problems.

ciated with 2 problems each. Below we present our observations on student difficulties (i.e. when and where in these structures students spent more time) associated with each problem structure:

Structure 1: In structure 1 [Figure 8a], the chunks are sequentially connected with different categories of rules. We observed that within each of the six difficult problems with this structure, there are almost no visual commonalities across the chunks. An example problem with this structure is shown in Figure 8d. In the figure, notice that each chunk contains propositions composed of variables from almost exclusive sets (Chunk 1 variables=S, I, Y, Q), Chunk 2 variables=D, Y), and also each chunk requires a different rule.

In these 6 problems, we found 71 students (T_1 = 39, T_2 = 32) who spent time above the 75th percentile to derive a chunk within at least one of these problems. A total of 247 difficult instances were found for these students solving problems with structure 1. 132 of those instances were associated with forward-directed sequential derivation (i.e., the students completed the problem in the following sequence, chunk 1 → chunk 2 → conclusion), 11 were associated with backward-directed sequential derivation (i.e., the students completed the problem in the following sequence, conclusion → chunk 2 → chunk 1), and 104 instances were associated with random derivation where students moved from chunk to chunk without demonstrating a strategical pattern.

Overall, after the analysis of the structure of the 6 problems and student approaches to solving the problems (forward, backward, or random), we could not associate a specific approach with the chunk-solving difficulty. Rather, we concluded that the difficulties could be associated with the diversity in rules/variables across chunks within the problems that possibly increased cognitive load[5] introducing difficulties for students.

Structure 2: In structure 2 [Figure 8b], two parallel chunks (chunk 1 and chunk 2) with very similar derivations are combined to derive the conclusion or a third chunk (chunk 3) that later helps to derive the conclusion. We found 2 difficult problems associated with structure 2. An example problem for this structure is shown in Figure 8e.

We identified 40 students (18 T_1 students, 22 T_2 students) who at least had one difficult instance (i.e. spent above 75th percentile of time) while solving one of the problems associated with this structure. 50 difficult instances [16 associated with forward-directed sequential derivation, 1 associated with backward-directed sequential derivation, and 33 with random derivation] were found for these students while solving one of these 2 problems. We observed that the students spent more time on either chunk 1 (31 instances) or chunk 2 (19 instances) depending on whichever they attempted to complete first. We also observed that they spent average or below-average time while deriving the rest of the chunks.

These observations indicate that the students were able to identify similarities across the chunks within a problem. Thus, although they spent more time on the first chunk, after figuring out the derivation of the first chunk, they needed less time to derive the rest.

Structure 3: Structure 2 and structure 3 [Figure 8c] are visually very similar. However, the main difference is that the derivations of chunk 1 and chunk 2 within structure 3 have no similarities (an example problem is shown in Figure 8f). We found 2 difficult problems associated with structure 3.

33 students were identified (19 T_1 students, 14 T_2 students) who had at least one difficult derivation (i.e. spent above 75th percentile of time) while solving one of the problems associated with this structure. 45 difficult instances [28 associated with FW-directed sequential derivation, 13 associated with BW-directed derivation, and 4 with random derivation] were found for these students while solving one of these 2 problems. And, we observed that in most of the cases, students spent higher time on both chunk 1 and chunk 2 (total 35 instances).

Overall, our observations indicate that difficulties mostly occurred when chunks within a problem were very dissimilar (in Structure 1 and Structure 3). On the other hand, if there are similar chunks within a problem, after deriving one chunk, the students figured out the derivation of other similar chunks very quickly.

4.2.3 *Learning Efficiency and Correlation Test between NLG and Training Time*

Overall, the training time for T_1 (this group received CPP without any explanation) and T_2 (this group received CPP with explanations) was higher than the control group [Control (C): 66.4(35.3) minutes, T_1 (CPP): 89.7 (60.7) minutes, T_2 (CPPE): 81.8 (46.0) minutes]. The skewed distribution of training times across the three training conditions is visualized in Appendix B, Figure 12.

Since the training times were higher for the treatment groups, we calculated the learning efficiency (NLG/Training Time) for each group. However, we did not find any difference in learning efficiency across the groups [Control (C): 0.007(0.011), T_1 (CPP): 0.003(0.005), T_2 (CPPE): 0.004(0.009). Kruskel-Wallis Test: (statistic=2.84, p-value=0.24); Pairwise post-hoc Mann Whitney U Tests: (C, T_1)=(statistic= 986.0, p-value=0.30), (C, T_2)=(statistic=1020.0, p-value = 0.11), (T_1, T_2)=(statistic=1231.0, p-value=0.43)]. We also did not find any significant correlation between NLG and training times [Control (C): coefficient = -0.09, p-value =

[5] The amount of working memory being used

0.32; T_1 (CPP): coefficient = -0.21, p-value = 0.96; T_2 (CPPE): coefficient = 0.01, p-value = 0.43]. Therefore, it is unclear whether or not the differences in NLG across the training conditions occurred due to differences in training times.

5. RQ3: STUDENTS' CHUNKING BEHAVIOR

CPP and CPPEs were incorporated within DT training levels to demonstrate chunking (i.e. decomposed problem solutions), engage students in the process (by having them recompose complete solutions from chunks), and motivate them to adopt chunking (i.e. decomposition-recomposition (PDR)) to reduce difficulties while solving new problems. To investigate if students successfully captured the notion of chunking while solving CPP/CPPE and if they tried to form chunks when solving problems independently (which we refer to as chunking behavior), we adopted a data-driven approach. From log data collected within DT, we tried to infer if the students showed chunking/PDR behavior and how the behavior was associated with their performance.

Method to Identify Chunking Behavior: We analyzed students' chunking behavior in the posttest problems that they solved independently [7.1-7.6]. To do so, first, we derived baseline chunks from historical student solutions to these problems. For problems 7.1-7.6, using the method described in Section 3.2, we generated approach maps using historical data collected in DT to capture previous students' solutions and identified baseline chunks in those solutions. The baseline chunks answer 'What to look for in the solutions of the students participating in this study'. Next, to confirm the presence of the baseline chunks or chunking/PDR behavior in a student's proof construction attempt, we sequentially scanned through the student's steps while constructing a proof and identified consecutive steps as chunking when the step sequence has the following characteristics:

1. The propositions derived in the step sequence overlaps with propositions and subgoal associated with only one of the baseline chunks [we applied the 'Intersection' set operation to find an overlap].

2. The step sequence may or may not be separated from the rest of the steps by a time gap above the average step time (the time spent on a single step) of 1.6 minutes. The two cases of separation by time gaps are shown in Sample 1 and Sample 2 in Figure 9.

The process of identifying chunking in a student solution attempt is further illustrated in Figure 9.

Learning and Chunking Behavior: Following prior research [9, 25], to validate our method to identify chunking behavior, we sought to explain students' learning gain that reflects their problem-solving skills through derivation efficiency in the chunking instances detected using our method.

We hypothesized that a higher number of treatment group students (who received CPP/CPPE) will have chunking instances in their solutions of posttest problems than the control(C) group students. However, we identified that most of the students (121 out of the 145 students) regardless of their training conditions had baseline chunks present in their solutions. Only 24 students (C=8,T_1=9,T_2=6) never showed any identifiable chunking behavior. This indicates that students might have a natural tendency to identify sub-problems and construct logic proofs in chunks. Whereas the presence of some chunking instances is desirable, too many chunks in a problem solution do not represent better performance and more learning. Deep Thought proofs are usually 7-15 steps long and an ideal solution for each problem mostly contains 2-3 chunks. Since the Deep Thought problem score which impacts NLG is designed as a function of time, step counts, and rule application accuracy, to achieve higher scores and NLG, students need to demonstrate only correct baseline chunks within their solutions and each of those chunks needs to be derived efficiently with less time and fewer steps. This fact was validated by a mediation analysis. In the analysis, we used training condition as the independent variable (IV), NLG as the dependent variable (DV), and average # chunks/problem as the mediator (MD). The analysis gave an insignificant p-value [Appendix B, Figure 14] indicating that the impact of the training treatments on learning or NLG is not mediated by the amount of chunking present in students' logic-proof solutions.

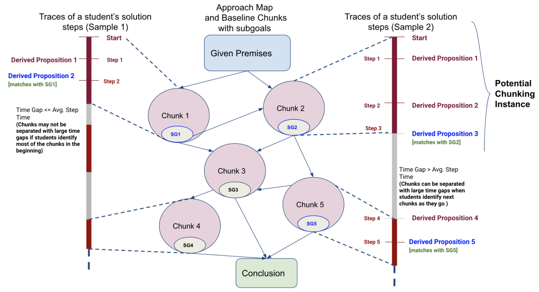

Traces of a student's solution steps (Sample 1)
Approach Map and Baseline Chunks with subgoals
Traces of a student's solution steps (Sample 2)
Start
Derived Proposition 1
Step 1
Derived Proposition 2
[matches with SG1]
Step 2
Time Gap <= Avg. Step Time
(Chunks may not be separated with large time gaps if students identify most of the chunks in the beginning)
Given Premises
Chunk 1
SG1
Chunk 2
SG2
Chunk 3
SG3
Chunk 4
SG4
Chunk 5
SG5
Conclusion
Start
Step 1
Derived Proposition 1
Step 2
Derived Proposition 2
Step 3
Derived Proposition 3
[matches with SG2]
Potential Chunking Instance
Time Gap > Avg. Step Time
(Chunks can be separated with large time gaps when students identify next chunks as they go)
Step 4
Derived Proposition 4
Step 5
Derived Proposition 5
[matches with SG5]

Figure 9: Method to Identify Chunking using Approach Map and Students' Solution Traces [showing the mapping between step sequence and chunks].

Thus, in subsequent analyses, instead of focusing on the number of chunks present in student solutions, we focused only on students' efficiency in deriving the baseline chunks. We calculated efficiency in terms of time spent on deriving different chunks and # of steps within the chunks. We also analyzed NLGs across different pretest score groups to understand the impact of CPP/CPPEs on students with different levels of prior knowledge.

Moderation Analysis on Different Pretest Score Groups: Prior studies showed that both worked examples and Parsons problems may have a different impact on students based on their prior knowledge or skill level [34, 17]. We carried out a moderation analysis to understand how NLG and chunking behavior and efficiency varied across different pretest score groups and training conditions. The pretest score distribution is visualized in Appendix B, Figure 13. We classified students based on their pretest scores (low, medium, and high) and used this classification as the moderator in our analysis. Within this classification, we considered training condition as the independent variable. For each pretest score group and training group, we analyzed 4 dependent variables: average # chunks/problem, average chunk time, avg. chunk step count, and the NLG. Table 2 shows the groupings and values for the dependent variables. To compare the dependent variables across pretest score groups and training conditions, we carried out Kruskal Wallis test and pairwise posthoc Mann Whitney U tests with Bonferroni correction (corrected $\alpha = 0.05/3$ or 0.016). Note that a total of 36 tests(3 pretest score groups, 4 dependent variables, and 3 pairwise tests for each group and dependent variable) were carried out to compare the metrics presented in Table

2. Thus, a more conservative Bonferroni correction could be carried out to eliminate false positives. However, to not introduce many false negatives while eliminating false positives, we decided the level of correction based on the number of unique pairwise tests each datapoint participated in [6] rather than on the number of related tests (for example, the 9 tests on NLG across the pretest scores groups though independent could be considered related and a more conservative correction could be carried out). We explain the results for each pretest score group below:

Low Pretest Scorers: We considered students with pretest scores below the 25th percentile as low scorers. In this group, we observed that T_2 who received CPPE had significantly higher and less negative NLGs than the other two training conditions [$p_{KW} < 0.001$, $p_{MW}(T_2 > C) = 0.011$, $p_{MW}(T_2 > T_1) < 0.001$]. We also observed that T_2 students with low pretest scores had lower average chunk time and significantly lower step counts per chunk [$p_{KW} < 0.03$, $p_{MW}(T_2 < C) = 0.004$, $p_{MW}(T_2 < T_1) < 0.014$]. Overall, T_2 students with low pretest scores demonstrated comparatively more efficient chunking (in terms of less time and step counts) and higher NLG. However, a difference in the number of chunks per problem was not found across the training conditions as expected.

Medium Scorers: Students with pretest scores between 25th-75th percentile were identified as the medium scorers. Within this group, we observed that the T_2 condition again showed significantly or marginally higher NLG than the other two training conditions [$p_{KW} < 0.001$, $p_{MW}(T_2 > C) = 0.002$, $p_{MW}(T_2 > T_1) < 0.021$]. We observed differences in the averages of chunk time across the three training conditions, however, a significant difference was not found in chunk count, time, or step counts.

High Scorers: We did not observe any significant differences in NLG across the three training conditions for students with high pretest scores (above 75th percentile). However, we observed that T_1 and T_2 students in the high pretest score group demonstrated comparatively more chunking (in the range of 3-4 chunks per problem) in posttest than the control (C) group (in the range of 2-3 chunks per problem).

Overall, the results of the moderation analysis indicate that T_2 students with low and medium pretest scores achieved significantly higher NLGs than students from the other two training conditions with similar levels of prior knowledge. There were no significant differences in the amount of chunking per problem across the training conditions. However, we observed differences in the chunking efficiency where T_2 had lower chunk derivation times and fewer steps within chunks in some cases. Thus, next, we analyze and present the chunking efficiency across the three training conditions on different posttest problems in further detail.

Chunk Derivation Efficiency: We compared the chunk derivation efficiency of the students across the three training conditions who had identifiable chunks in their solutions. Toward that, we analyzed two metrics for the baseline chunks identified in student solutions to the posttest problems: 1) Time to derive a chunk (shorter chunk derivation time [CTime] indicates students figured out '*how to derive the chunk*' quickly), 2) unnecessary proposition count [UProp][6] (fewer unnecessary propositions indicate students correctly identified '*what to derive within the chunk*'). Lower values for these two metrics indicate higher chunk derivation efficiency.

In Figure 10, we show the chunks commonly found in students' proof for each of the posttest problems. For simplicity, for each of the chunks, we only show what subgoal the chunk derives. To identify significant differences in the derivation efficiency of these chunks (in terms of CTime or UProp) across the three training conditions, we carried out Kruskal-Wallis tests. The chunks for which there is a significant difference in derivation efficiency across the training conditions in terms of at least one of UProp or CTime are marked with thicker edges and green nodes in the figure. The results of the statistical tests are shown along the thicker edges. We observed that the significant differences were found mostly for non-trivial chunks, i.e. chunks that involve several proposition derivations. For example, there are two chunks in the solution of 7.1: the first chunk derives $\neg R$ which requires multiple steps (i.e. non-trivial), and the second chunk derives $R \vee \neg T$ which can be derived after a *Simplification* rule application on the given premise $(R \vee \neg T) \wedge X$ (trivial derivation). We found significant differences only in the derivation efficiency of chunk 1 (the non-trivial chunk). To identify the training condition that was the most efficient in deriving the green chunks in Figure 10, we carried out posthoc pairwise Mann-Whitney U tests [for the pairs (C, T_1), (C, T_2), and (T_1, T_2)] with Bonferroni correction (corrected α=0.016) comparing UProb and CTime. The results of the tests are shown in Table 3. As shown in the table, in most of the cases, T_2 is the most efficient group in deriving the chunks, i.e. the tests for the hypotheses '$T_2 < C$' and '$T_2 < T_1$' in terms of UProp/CTime gave p-value < 0.016. Overall, these results indicate that although most students naturally derived chunks, T_2 students achieved higher efficiency in deriving non-trivial chunks.

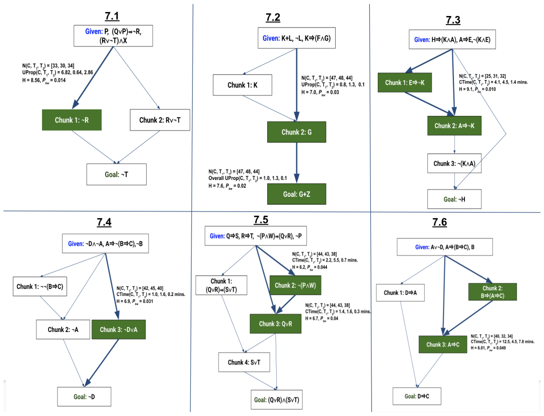

7.1
Given: P, (Q∨P)⇒¬R, (R∨¬T)∧X
N(C, T1, T2) = [33, 30, 34]
UProp(C, T1, T2) = 6.82, 0.64, 2.86
H = 8.56, Pkw = 0.014
Chunk 1: ¬R
Chunk 2: R∨¬T
Goal: ¬T
7.2
Given: K+L, ¬L, K⇒(F∧G)
Chunk 1: K
N(C, T1, T2) = [47, 48, 44]
UProp(C, T1, T2) = 0.8, 1.3, 0.1
H = 7.0, Pkw = 0.03
Chunk 2: G
N(C, T1, T1) = [47, 48, 44]
Overall UProp(C, T1, T2) = 1.0, 1.3, 0.1
H = 7.6, Pkw = 0.02
Goal: G+Z
7.3
Given: H⇒(K∧A), A⇒E,¬(K∧E)
Chunk 1: E⇒¬K
N(C, T1, T2) = [25, 31, 32]
CTime(C, T1, T2) = 4.1, 4.5, 1.4 mins.
H = 9.1, Pkw = 0.010
Chunk 2: A⇒¬K
Chunk 3: ¬(K∧A)
Goal: ¬H
7.4
Given: ¬D∧¬A, A⇒¬(B⇒C),¬B
Chunk 1: ¬¬(B⇒C)
N(C, T1, T2) = [42, 45, 40]
CTime(C, T1, T2) = 1.0, 1.6, 0.2 mins.
H = 6.9, Pkw = 0.031
Chunk 2: ¬A
Chunk 3: ¬D∨A
Goal: ¬D
7.5
Given: Q⇒S, R⇒T, ¬(P∧W)⇒(Q∨R), ¬P
N(C, T1, T2) = [44, 43, 38]
CTime(C, T1, T2) = 2.2, 5.5, 0.7 mins.
H = 6.2, Pkw = 0.044
Chunk 1: (Q∨R)⇒(S∨T)
Chunk 2: ¬(P∧W)
N(C, T1, T2) = [44, 43, 38]
CTime(C, T1, T2) = 1.4, 1.6, 0.3 mins.
H = 6.7, Pkw = 0.04
Chunk 3: Q∨R
Chunk 4: S∨T
Goal: (Q∨R)∧(S∨T)
7.6
Given: A∨¬D, A⇒(B⇒C), B
Chunk 1: D⇒A
Chunk 2: B⇒(A⇒C)
Chunk 3: A⇒C
N(C, T1, T2) = [40, 32, 34]
CTime(C, T1, T2) = 12.5, 4.5, 7.8 mins.
H = 6.01, Pkw = 0.049
Goal: D⇒C

Figure 10: Chunk Derivation Efficiency in Posttest Problems.

6. DISCUSSION

Overall, our analysis showed that *Chunky Parsons problem* could be a low-difficulty training intervention, specifically when presented with an explanation hinting at what the chunks mean and how they contribute to the complete solution. However, while being a low-difficulty training intervention, it has the potential to improve students' learning gain and problem-solving skills, specifically chunking skills. We observed that most students formed some chunks dur-

[6] Unnecessary propositions are propositions that students derived during proof construction but later deleted and those were not part of the final proof.

Table 2: Moderation Analysis across the Three Training Conditions Categorized on Pretest Scores. [Note: Blue* indicates a significant difference. Boldface indicates comparatively better averages (e.g. higher for NLG/lower for extra steps).]

Moderator	Independent Variable (IV)	Dependent Variables (DV)			
Pretest Quantile	Training Condition	Avg. # chunks/ prob.	NLG	Avg. Chunk Time (minutes)	Avg. Chunk Step Count
Low Scorers (< 25th percentile) N = 35	Control(n=12)	2.02(3.25)	-0.20(0.35)	3.42(1.45)	2.01(0.81)
	T_1-CPP(n=12)	2.47(3.62)	-0.40(0.39)	2.78(3.80)	2.06(1.03)
	T_2-CPPE(n=11)	2.19(3.54)	**-0.07(0.22)***	**2.21(3.26)**	**1.94(0.90)***
Medium Scorers (25th-75th percentile) N = 75	Control(n=25)	2.51(3.98)	0.34(0.23)	2.88(4.51)	2.25(1.38)
	T_1-CPP(n=26)	2.58(3.96)	0.35(0.25)	3.03(5.15)	2.05(1.08)
	T_2-CPPE(n=23)	2.26(3.60)	**0.48(0.22)***	**2.45(3.90)**	2.14(1.14)
High Scorers (> 75th percentile) N = 35	Control(n=13)	2.92(3.72)	0.30(0.17)	3.09(4.22)	2.45(1.34)
	T_1-CPP(n=11)	**3.50(4.5)**	0.32(0.13)	4.03(5.64)	2.16(0.89)
	T_2-CPPE(n=11)	**3.13(3.96)**	0.33(0.17)	**2.94(3.47)**	2.31(1.26)

Table 3: Chunk Derivation Efficiency across the Three Training Groups (only significant p-values are shown).

Problem	Chunk	Metric	Pairwise Mann-Whitney U Test
7.1	Chunk 1	UProp	p(T1<C)=0.005 p(T2<C)=0.012
7.2	Chunk 2	UProp	p(T2<C)=0.016 p(T2<T1)=0.006
7.3	Chunk 1 + Chunk 2	CTime	p(T2<C)=0.015 p(T2<T1)=0.002
7.4	Chunk 3	CTime	p(T2<C)=0.013 p(T2<T1)=0.014
7.5	Chunk 2 + Chunk 3	CTime	p(T2<C)=0.010 p(T2<T1)=0.030
7.6	Chunk 2 + Chunk 3	CTime	p(T1<C)=0.010 p(T2<C)=0.020

ing proof construction. However, students from all training conditions were not equally efficient in chunking. Our statistical tests showed that T_2 (who received CPP with an explanation attached to chunks) derived non-trivial chunks with higher efficiency. However, this efficiency often was not observed for all chunks within a problem. Another limitation of CPP/CPPEs is the difficulties associated with it when students first encounter CPP/CPPE in early training levels or when the chunks within a CPP/CPPE are very diverse. Thus, we recommend providing additional guidance or tutor help in these scenarios to ensure a better student experience while solving and learning through CPP/CPPE. Nevertheless, our analyses have established that Chunky Parsons Problems with explanations can be an effective problem-based training intervention to improve students' Chunking skills (i.e. problem decomposition into chunks and recomposing them to construct a complete solution).

Also, our data-driven method to derive subgoals can be adopted for any structured problem-solving domain as long as each step during problem solving can be presented as a state transition. For example, in a math-expression evaluation problem, a state can be the set of all evaluated parts of the equation at a particular moment. A step or an action (for example, applying a math operator) changes the problem state. Once the state transitions or interaction is defined within a domain, generating the approach maps and extracting subgoals from them can be carried out generically (graph construction, applying clustering, and simplifying the graph in approach maps). Similarly, the chunking efficiency evaluation method can be adopted in other domains, as long as each point in the students' sequential problem solution traces can be presented as a state from a finite state space.

7. CONCLUSION AND FUTURE WORK

The contributions of this paper are 1) the demonstration of a data-driven graph-mining-based method to decompose problem solutions into expert-level chunks, 2) the design of a problem-based training intervention called *Chunky Parsons Problem* to be used within an intelligent tutor to teach students the concept of structural decomposition-recomposition (or Chunking) of problems, 3) an evaluation of the impact of *Chunky Parsons Problem* on learning and students' chunking skills, and 4) a mechanism to identify *Chunking* in students' solution traces using historical baseline chunks. As discussed earlier, our data-driven methods to derive *Chunky Parsons Problem* and to identify *Chunking* in student solution traces can be adapted for any domain where problem solving is structured and the states and transitions of students during problem solving can be defined definitely. Likewise, *Chunky Parsons Problem* can be adapted for any problem-based tutor within such domains.

However, this study has several limitations. First, the design decisions for Chunky Parsons Problems (CPPs) and their explanations were made based on prior literature, without any user studies to validate them. Second, while we validated chunks found in participants' solutions, our data-driven evaluation method may not be able to detect new chunks that were not previously seen in prior student data. Also, the outcomes of this study are dependent on how we defined different data-driven metrics (for example, difficulty or efficiency). Third, although our evaluation can identify the impact of interventions, it cannot validate the source of the impact. Thus, future user studies involving interviews or talk-aloud protocols could help address these three issues and validate the findings on the usability and impact of *Chunky Parsons Problem*. Finally, our study focused only on logic-proof problems and should be replicated in other domains to understand the generalizability of the findings.

8. ACKNOWLEDGMENTS

The work is supported by NSF grant 2013502.

9. REFERENCES

[1] M. Abdelshiheed, J. W. Hostetter, P. Shabrina, T. Barnes, and M. Chi. The power of nudging: Exploring three interventions for metacognitive skills instruction across intelligent tutoring systems. In *Proceedings of the 44th annual conference of the cognitive science society*, pages 541–548, 2022.

[2] M. Abdelshiheed, J. W. Hostetter, X. Yang, T. Barnes, and M. Chi. Mixing backward-with forward-chaining for metacognitive skill acquisition and transfer. In *Artificial Intelligence in Education*, pages 546–552. Springer, 2022.

[3] M. Abdelshiheed, G. Zhou, M. Maniktala, T. Barnes, and M. Chi. Metacognition and motivation: The role of time-awareness in preparation for future learning. In *Proceedings of the 42nd annual conference of the cognitive science society*, pages 945–951, 2020.

[4] T. Barnes and J. Stamper. Toward the extraction of production rules for solving logic proofs. In *AIED07, 13th International Conference on Artificial Intelligence in Education, Educational Data Mining Workshop*, pages 11–20, 2007.

[5] D. Barr, J. Harrison, and L. Conery. Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, 38(6):20–23, 2011.

[6] J. M. Bland and D. G. Altman. Multiple significance tests: the bonferroni method. *Bmj*, 310(6973):170, 1995.

[7] R. Catrambone. Aiding subgoal learning: Effects on transfer. *Journal of educational psychology*, 87(1):5, 1995.

[8] R. Catrambone. The subgoal learning model: Creating better examples so that students can solve novel problems. *Journal of experimental psychology: General*, 127(4):355, 1998.

[9] C. Charitsis, C. Piech, and J. C. Mitchell. Using nlp to quantify program decomposition in cs1. In *Proceedings of the Ninth ACM Conference on Learning@ Scale*, pages 113–120, 2022.

[10] O. Chen, S. Kalyuga, and J. Sweller. The worked example effect, the generation effect, and element interactivity. *Journal of Educational Psychology*, 107(3):689, 2015.

[11] T. J. Cortina. An introduction to computer science for non-majors using principles of computation. *Acm sigcse bulletin*, 39(1):218–222, 2007.

[12] G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations research*, 8(1):101–111, 1960.

[13] P. J. Denning. The profession of it beyond computational thinking. *Communications of the ACM*, 52(6):28–30, 2009.

[14] P. Denny, A. Luxton-Reilly, and B. Simon. Evaluating a new exam question: Parsons problems. In *Proceedings of the fourth international workshop on computing education research*, pages 113–124, 2008.

[15] M. Eagle and T. Barnes. Exploring differences in problem solving with data-driven approach maps. In *Educational data mining 2014*, 2014.

[16] M. Eagle, D. Hicks, and T. Barnes. Interaction network estimation: Predicting problem-solving diversity in interactive environments. *International Educational Data Mining Society*, 2015.

[17] B. J. Ericson, J. D. Foley, and J. Rick. Evaluating the efficiency and effectiveness of adaptive parsons problems. In *Proceedings of the 2018 ACM Conference on International Computing Education Research*, pages 60–68, 2018.

[18] G. V. F. Fabic, A. Mitrovic, and K. Neshatian. Evaluation of parsons problems with menu-based self-explanation prompts in a mobile python tutor. *International Journal of Artificial Intelligence in Education*, 29(4):507–535, 2019.

[19] K. B. Gallagher and J. R. Lyle. Using program slicing in software maintenance. *IEEE transactions on software engineering*, 17(8):751–761, 1991.

[20] J. S. Gero and T. Song. The decomposition/recomposition design behavior of student and professional engineers. In *2017 ASEE Annual Conference & Exposition*, 2017.

[21] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[22] S. Grover and R. Pea. Computational thinking in k–12: A review of the state of the field. *Educational researcher*, 42(1):38–43, 2013.

[23] M. Guzdial. Education paving the way for computational thinking. *Communications of the ACM*, 51(8):25–27, 2008.

[24] V. Karavirta, J. Helminen, and P. Ihantola. A mobile learning application for parsons problems with automatic feedback. In *Proceedings of the 12th koli calling international conference on computing education research*, pages 11–18, 2012.

[25] J. S. Kinnebrew, K. M. Loretz, and G. Biswas. A contextualized, differential sequence mining method to derive students' learning behavior patterns. *Journal of Educational Data Mining*, 5(1):190–219, 2013.

[26] K. Kwon and J. Cheon. Exploring problem decomposition and program development through block-based programs. *International Journal of Computer Science Education in Schools*, 3(1):n1, 2019.

[27] L. A. Liikkanen and M. Perttula. Exploring problem decomposition in conceptual design among novice designers. *Design studies*, 30(1):38–59, 2009.

[28] M. Maniktala and T. Barnes. Deep thought: An intelligent logic tutor for discrete math. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, pages 1418–1418, 2020.

[29] L. E. Margulieux, M. Guzdial, and R. Catrambone. Subgoal-labeled instructional material improves performance and transfer in learning to develop mobile applications. In *Proceedings of the ninth annual international conference on International computing education research*, pages 71–78, 2012.

[30] J. J. McConnell and D. T. Burhans. The evolution of cs1 textbooks. In *32nd Annual Frontiers in Education*, volume 1, pages T4G–T4G. IEEE, 2002.

[31] B. B. Morrison, L. E. Margulieux, B. Ericson, and M. Guzdial. Subgoals help students solve parsons problems. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pages 42–47, 2016.

[32] O. Muller, D. Ginat, and B. Haberman. Pattern-oriented instruction and its influence on problem decomposition and solution construction. In *Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education*, pages 151–155, 2007.

[33] G. B. Mund and R. Mall. An efficient interprocedural dynamic slicing method. *Journal of Systems and Software*, 79(6):791–806, 2006.

[34] F. Nievelstein, T. Van Gog, G. Van Dijck, and H. P. Boshuizen. The worked example and expertise reversal effect in less structured tasks: Learning to reason about legal cases. *Contemporary Educational Psychology*, 38(2):118–125, 2013.

[35] F. Paas, A. Renkl, and J. Sweller. Cognitive load theory and instructional design: Recent developments. *Educational psychologist*, 38(1):1–4, 2003.

[36] J. L. Pearce, M. Nakazawa, and S. Heggen. Improving problem decomposition ability in cs1 through explicit guided inquiry-based instruction. *J. Comput. Sci. Coll*, 31(2):135–144, 2015.

[37] C. Piech, M. Sahami, D. Koller, S. Cooper, and P. Blikstein. Modeling how students learn to program. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, pages 153–160, 2012.

[38] S. Poulsen, M. Viswanathan, G. L. Herman, and M. West. Evaluating proof blocks problems as exam questions. *ACM Inroads*, 13(1):41–51, 2022.

[39] A. Renkl. The worked-out-example principle in multimedia learning. *The Cambridge handbook of multimedia learning*, pages 229–245, 2005.

[40] W. J. Rijke, L. Bollen, T. H. Eysink, and J. L. Tolboom. Computational thinking in primary school: An examination of abstraction and decomposition in different age groups. *Informatics in education*, 17(1):77–92, 2018.

[41] T. Song and K. Becker. Expert vs. novice: Problem decomposition/recomposition in engineering design. In *2014 International Conference on Interactive Collaborative Learning (ICL)*, pages 181–190. IEEE, 2014.

[42] R. Sooriamurthi. Introducing abstraction and decomposition to novice programmers. *ACM SIGCSE Bulletin*, 41(3):196–200, 2009.

[43] J. Stamper, T. Barnes, and M. Croy. Enhancing the automatic generation of hints with expert seeding. *International Journal of Artificial Intelligence in Education*, 21(1-2):153–167, 2011.

[44] J. Sweller and G. A. Cooper. The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and instruction*, 2(1):59–89, 1985.

[45] Y. Tao and S. Kim. Partitioning composite code changes to facilitate code review. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, pages 180–190. IEEE, 2015.

[46] P. Tonella. Using a concept lattice of decomposition slices for program understanding and impact analysis. *IEEE transactions on software engineering*, 29(6):495–509, 2003.

[47] N. Tsantalis and A. Chatzigeorgiou. Identification of extract method refactoring opportunities for the decomposition of methods. *Journal of Systems and Software*, 84(10):1757–1782, 2011.

[48] N. Weinman, A. Fox, and M. A. Hearst. Improving instruction of programming patterns with faded parsons problems. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–4, 2021.

[49] R. M. Young. Surrogates and mappings: Two kinds of conceptual models for interactive devices. In *Mental models*, pages 43–60. Psychology Press, 2014.

[50] S. Zhang, B. G. Ryder, and W. Landi. Program decomposition for pointer aliasing: A step toward practical analyses. In *Proceedings of the 4th ACM SIGSOFT Symposium on Foundations of Software Engineering*, pages 81–92, 1996.

[51] R. Zhi, M. Chi, T. Barnes, and T. W. Price. Evaluating the effectiveness of parsons problems for block-based programming. In *Proceedings of the 2019 ACM Conference on International Computing Education Research*, pages 51–59, 2019.

APPENDIX

A. DEEP THOUGHT INTERFACE AND FUNCTIONALITIES

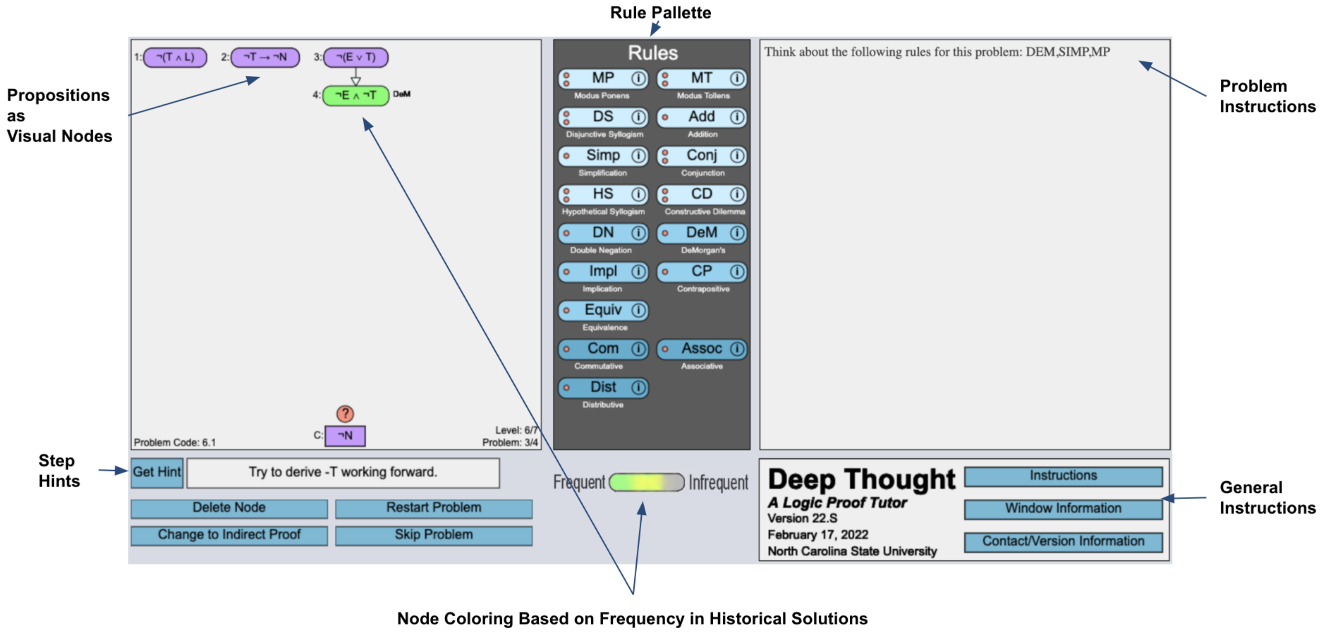
Figure 11 shows the different components and architecture of Deep Thought or DT, including one property worth mentioning: during training problems, the tutor colors student-derived propositions based on their frequency in prior student solutions. This coloring is designed to help the students to understand if they are on the right track or not.

A.1 Problem-Solving Strategies within Deep Thought

Logic Proof Construction problems in Deep Thought can be constructed using one of the three following strategies: 1) Forward Problem Solving; 2) Backward Problem Solving; and 3) Indirect Problem Solving. The three strategies are briefly described below:

Forward Problem Solving: In this strategy (Figure 15a), proof construction progresses from the given premises toward the conclusion. At each step, a new node is derived by applying rules on the given premises or derived justified nodes. To derive a new node in the forward direction, students first need to select the correct number of premise(s) or already justified node(s) and then select the rule to apply to the selected node(s).

Backward Problem Solving: In this strategy (Figure 15b), proof construction progresses from the conclusion toward the given premises. At each step, the conclusion is refined to a new goal. In this strategy, students can add unjustified nodes in the proof that they wish to derive from the given premises. To derive a node backward, students need to select the '?' button above a node, then select the rule, and then input the proposition(s) which are the antecedents of the selected node as per the selected rule. For example, in Figure 15b, the conclusion $\neg N$ is first refined into antecedents $\neg T \rightarrow \neg N$ and $\neg T$ using the Modus Ponens (MP) rule. $\neg T \rightarrow \neg N$ (given) is already justified. So, $\neg T$ becomes the new goal since it is still unjustified. Then, $\neg T$ is refined



Think about the following rules for this problem: DEM,SIMP,MP

Try to derive -T working forward.

Figure 11: Deep Thought Interface

to a new goal $\neg E \vee \neg T$ using the Simplification (Simp) rule. In this way, the unjustified goal(s) are refined to the given premises to complete the proof.

Indirect Problem Solving: Indirect problem solving [2] refers to the 'Proof by contradiction' approach. To construct a proof using this strategy, students first need to click on the 'Change to Indirect Proof' button (Figure 11) which adds the negation of the original conclusion to the list of givens (as in $\neg\neg N$ in Figure 15c). From there, students need to derive two contradictory statements (for example, $\neg\neg N$ and $\neg N$) to prove the contradiction ($\emptyset$).

Note that usually within Deep Thought, students are not required to follow any particular strategy. They can use any strategy at any point in a proof construction attempt.

B. SUPPLEMENTARY FIGURES

Figure 12, 13, and 14 supplement the analyses presented in Section 4.2.3 and 5.

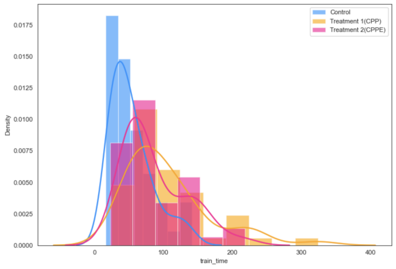


Figure 12: Training Time across the Three Training Conditions.

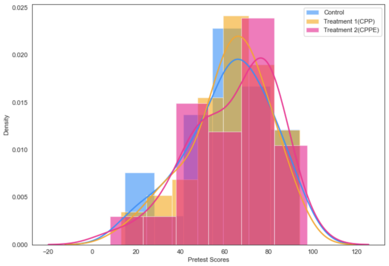


Figure 13: Distribution of Pretest Scores across the Three Training Conditions.

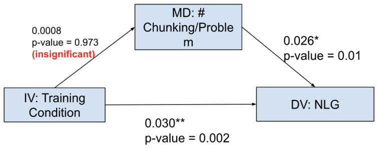
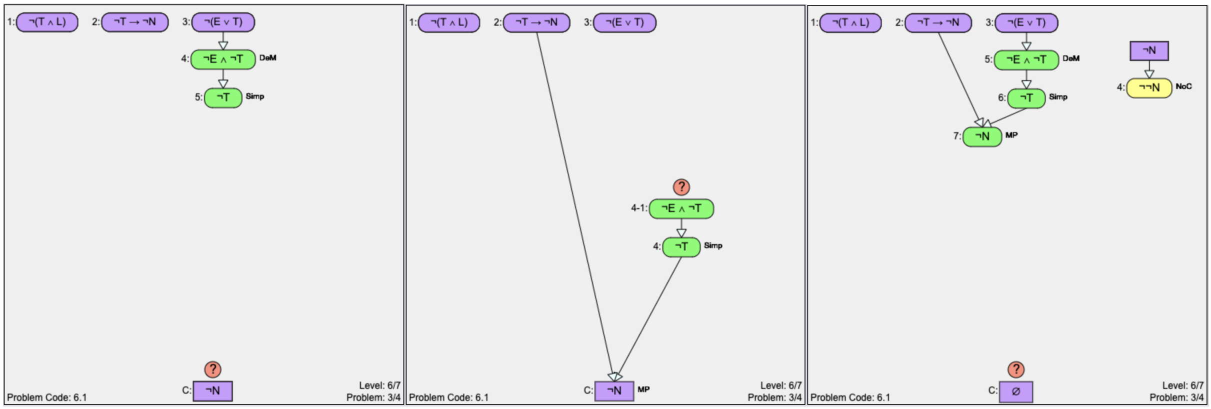


Figure 14: Mediation Analysis to Analyze the Impact of Amount of Chunking on Learning.



(a) Forward Strategy (b) Backward Strategy (c) Indirect Strategy

Figure 15: Problem-Solving Strategies Implementable in Deep Thought

An Analysis of Diffusion of Teacher-curated Resources on Pinterest

Hamid Karimi
Utah State University
hamid.karimi@usu.edu

Kaitlin Torphy Knake
Michigan State University
torphyka@msu.edu

Kenneth A. Frank
Michigan State University
kenfrank@msu.edu

ABSTRACT

Teachers increasingly rely on online social media platforms to supplement their educational resources, greatly influencing PK-12 education through the swift and extensive diffusion of teacher-curated resources. Understanding this diffusion process is crucial, but current educational studies primarily report resource diffusion through small-scale analyses, such as teacher interviews or anecdotal accounts. To bridge this gap, we conduct a pioneering, large-scale, quantitative, and data-driven analysis of the diffusion of teacher-curated resources on Pinterest, a platform widely embraced by educators. Our study begins by defining a resource's diffusion tree, which encapsulates the cascade of resource sharing across the social network. Based on this diffusion tree, we identify three measures to characterize a resource's diffusion process: volume, virality, and velocity. Equipped with these three measures, we conduct an in-depth analysis of the diffusion of over one million resources curated by thousands of teachers on Pinterest. Our investigation concludes by examining the correlation between a resource's attributes and its curator's attributes and the diffusion of the resource.

Keywords

Teachers, Social Media, Diffusion, Pinterest, Education

1. INTRODUCTION

Historically, teachers expanded their knowledge base through formal and informal professional development channels. They formed networks through direct, face-to-face interactions with peers and, more recently, through online communities for exchanging knowledge, experiences, and social capital. The emergence of online social media platforms like Facebook, Twitter, and Pinterest has provided teachers with a new platform for connecting with like-minded peers and sharing pedagogical resources. This phenomenon has stimulated a surge in academic studies focused on teachers' engagement with social media [1, 2, 3, 4, 5, 6]. Contrasting traditional methods of educational resource curation, which can be time-consuming and scale-limited, the accessibility of sourcing educational resources from fellow teachers on social media platforms has become highly appealing. Teachers can now readily access materials from those they admire or perceive as field experts.

Additionally, the diffusion of these resources can occur swiftly, often within the same day, enabling teachers to integrate new materials into their classroom practices efficiently. Across social media, the established social networks and professional communities of teachers have facilitated the diffusion of information and instructional resources on an unprecedented scale [7]. Previously, teachers might have had only a handful of colleagues to turn to for advice or information. Now, they can access a broad spectrum of instructional resources and interact with "teacherpreneurs" from across the globe [8]. Consequently, the fast and efficient diffusion of resources has become a new norm, significantly influencing pedagogical practices and educational dynamics.

While previous research has explored the diffusion of information among teachers, often referred to as the exchange of knowledge or resources [9], there remains to be a significant gap in our understanding of the large-scale propagation of teacher-curated resources on social media platforms. Specifically, investigations need to be more into how these resources navigate through the network and the influence of the resource attributes and its curator on this propagation process.

To address this, we conduct a comprehensive, large-scale analysis of the diffusion of teacher-curated resources on Pinterest, a platform popular among teachers [10]. We start by gathering a substantial sample of Pinterest-using teachers and detailed information about their curated resources. Subsequently, we construct the diffusion process for over one million teacher-curated resources on Pinterest. This process encapsulates several vital elements: the initial curator of a resource, subsequent users who have re-shared the resource, and the timeline of the resource's re-sharing.

These vital details about a resource's diffusion process are captured in a diffusion tree, as demonstrated in Figure 1, where we also display the pin curation time beneath each node (more about diffusion trees in Section 3.3). This example illustrates the speed of resource diffusion via social media, highlighting the platform's power in swiftly dissemi-

H. Karimi, K. T. Knake, and K. A. Frank. An analysis of diffusion of teacher-curated resources on pinterest. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 55–66, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115713

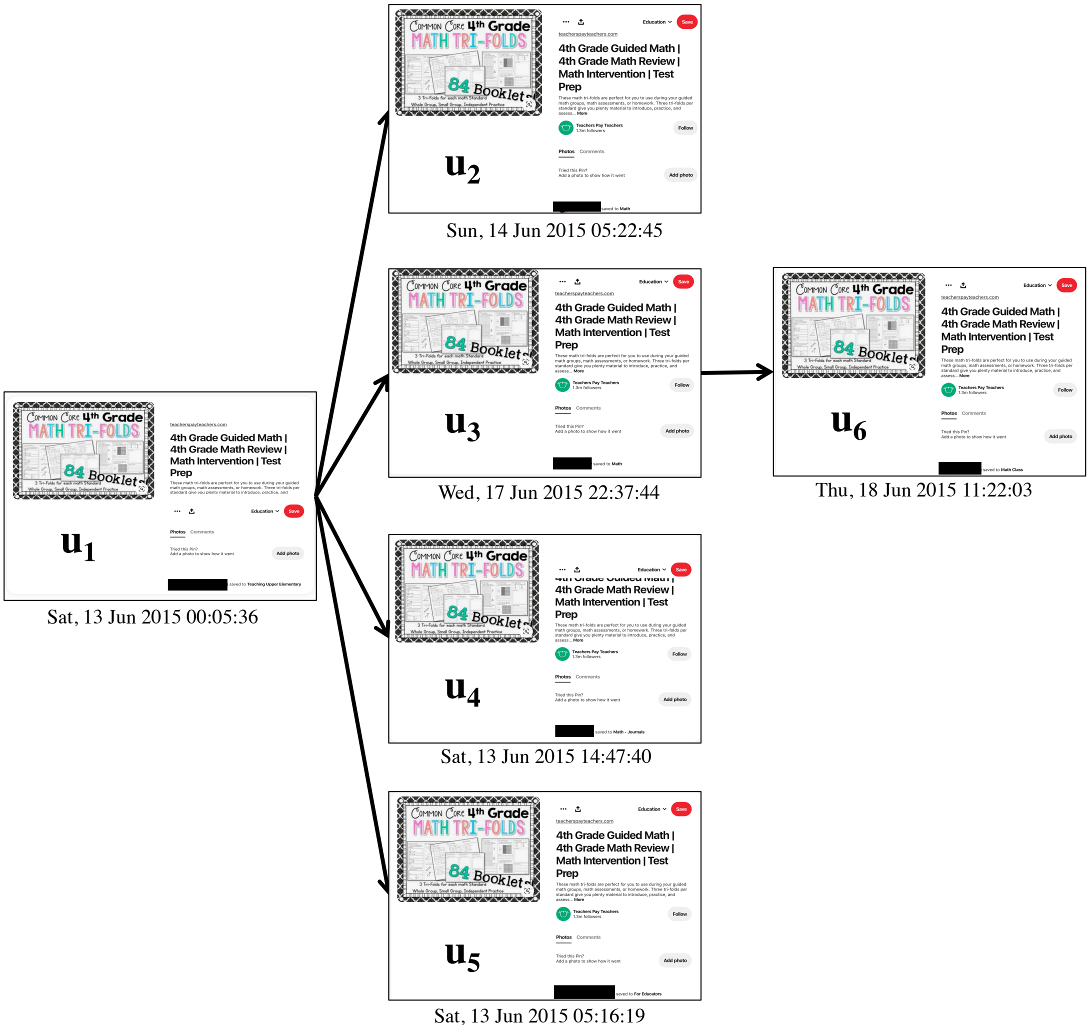

u2
Sun, 14 Jun 2015 05:22:45
u3
Wed, 17 Jun 2015 22:37:44
u6
Thu, 18 Jun 2015 11:22:03
u1
Sat, 13 Jun 2015 00:05:36
u4
Sat, 13 Jun 2015 14:47:40
u5
Sat, 13 Jun 2015 05:16:19

Figure 1: An example of a diffusion tree illustrating the prorogation of a teacher-curated resource on Pinterest

nating educational resources.

We then introduce three key measures that characterize the diffusion process: the number of users who have received a teacher-curated resource (volume), the structure-related penetration of a resource in the network (virality), and the speed of resource diffusion (velocity). Leveraging these measures, we conduct a large-scale analysis of teacher-curated resource diffusion and address two pivotal research questions. First, do resource attributes, such as their topics or sources, impact diffusion? Second, how do teacher-related attributes, such as the number of online followers, influence the diffusion of their curated resources?

This study's novel analysis and findings significantly contribute to the knowledge surrounding teaching and teacher learning with social media. Specifically, it helps illuminate how social media assists teachers in acquiring resources for their pedagogical practices. In summary, our contributions in this study are as follows:

- ➤ We construct the diffusion trees for over one million resources shared by thousands of teachers on Pinterest, offering a comprehensive visualization of resource propagation.
- ➤ We introduce three key measures - volume, virality, and velocity - to effectively characterize the diffusion of resources based on the constructed diffusion trees.
- ➤ We conduct a large-scale analysis of the diffusion of teacher-curated resources, outlining the relationship between the attributes of a resource and a teacher and how these relate to resource diffusion.

The rest of this paper is organized as follows. First, in Section 2, we present a brief literature review. Next, in Section 3, we discuss the dataset. Then, in Section 4, we introduce measures characterizing the diffusion process. Section 5 includes our analysis of the diffusion of teacher-curated resources. Finally, we conclude the paper in Section 6.

2. RELATED WORK

Online social media platforms offer significant benefits to teachers, notably in the domain of instructional resource curation [3]. Pinterest, an image-based personalized social media platform, is pivotal in this regard, boasting 440 million active users per month [11]. American teachers widely adopt it as a professional platform and a virtual repository of resources [8, 12, 13]. A national survey by the RAND Corporation underscores this trend, revealing that most elementary and secondary teachers in the United States utilize Pinterest to cater to their instructional needs [10].

The qualitative analysis conducted by the authors in [13], based on interviews with eight teachers, sheds light on the functionality of Pinterest in the educational sphere. They recruited teachers through snowball sampling on Twitter and found that educators viewed Pinterest as a digital organizer, compiling resources they discovered online or developed themselves. This echoes findings from previous studies that emphasized Pinterest's role as a content curation tool [14, 15, 16, 17].

Further exploring this theme, Schroeder et al. [18] conducted a qualitative study involving 117 teachers and found that educators predominantly used Pinterest to find resources tailored to their classroom requirements. Moreover, Torphy Knake et al. [8] investigated teacherpreneurial behaviors on Pinterest. After analyzing the source of 140,287 resources curated by 197 teachers, they found that educational blogs were the primary origin of these resources. In addition, market websites specifically targeting teachers, notably teacherspayteachers.com, also contributed significantly to the source of pins.

Additionally, their study revealed that a substantial majority of pins (82.8%) were monetized. Hu et al. [12] examined the curation mechanism of mathematical resources on Pinterest, discovering that these resources typically exhibited low cognitive demand. Their research also demonstrated the role of socialized knowledge communities in assisting mathematics teachers in finding relevant resources. Lastly, [19] provided insightful analysis into the curation practices of mathematical resources, identifying three types: self-directed, incidental, and socialized. A key takeaway from their study is their insight into how Pinterest-sourced educational resources are utilized in the classroom.

The work most closely related to our study is that by Liu et al. [20], which examined the process of Pinterest resource curation among 34 early career teachers (ECTs) from three states in the Midwest. They focused on the diffusion of resources among an ECT and their colleagues within the same school, whom the ECT nominated as close colleagues. Their findings suggest that Pinterest serves as a conduit between weakly connected teachers in the same school.

However, our study presents several significant improvements compared to [20]. Firstly, we operate on a much larger scale, investigating the diffusion of over one million resources among thousands of teachers. Secondly, while their study examined diffusion through a single direct re-pinning between two teachers, we delve into the entire cascade of information diffusion as represented by the diffusion trees. Thirdly, their study was limited to teachers within the same school who have potential face-to-face interactions. In contrast, we examine diffusion among teachers on an online platform without consideration for potential real-life interactions.

3. DATASET

This section provides an overview of the dataset we utilized for our study. We detail the process of teacher sampling, explain our approach to automatic teacher identification, and illustrate how we construct the diffusion trees.

3.1 Teacher Sampling

As a part of an interdisciplinary project called "Teachers in Social Media"[1], we surveyed 540 teachers across five U.S. states: Illinois, Indiana, Michigan, Ohio, and Texas. We then harnessed the Pinterest API (Application Programming Interface) to gather data about these surveyed teachers and their online connections, including followers and followees. The collected data for each user encompasses their pins and boards. Every pin carries an image (or, in recent times, a video), a description, a title, a link to its source, a board, the parent pin, and other supplementary information. The parent pin refers to the preceding pin from which the current pin has been re-pinned (re-shared). A board is a user-generated catalog that organizes pins with similar themes (for instance, all pins related to 'multiplication table instruction').

3.2 Automatic Teacher Identification

As stated earlier, the principal aim of this paper is to conduct a large-scale analysis of the diffusion of resources curated by teachers. However, utilizing data from only the surveyed teachers would not suffice to accomplish this goal, as we have surveyed a relatively small number of teachers. Therefore, one might suggest increasing the number of surveyed teachers. However, surveying is a time-consuming and expensive process. As such, developing a method capable of identifying teachers *automatically* becomes highly beneficial, especially considering that we have already collected data from thousands of users connected to the surveyed teachers. Moreover, based on the principle of homophily (i.e., the tendency for individuals to associate with others similar to themselves [21]), which is prevalent in (online) social networks, it is highly probable that a significant portion of the surveyed teachers' online connections are indeed teachers.

Fortunately, in our prior study [5], we introduced a machine learning-based method capable of efficiently and effectively identifying teachers on Pinterest. For reference, Figure 2 provides a comprehensive view of our previously proposed method. The input for this method is the data of an unlabeled user (i.e., an online friend of a surveyed teacher), and the output is the probability that this user is a teacher, denoted as p. We establish a threshold τ; if $p > \tau$, the user is considered a teacher; otherwise, they are classified as a non-teacher. Employing a conservative threshold of $\tau = 0.9$, we automatically identified approximately 16,000 additional teachers. Our rigorous evaluation of this method in our previous study indicated a minimal error in teacher classification. Specifically, we conducted an exhaustive resiliency

[1] `https://www.teachersinsocialmedia.com/`

analysis of this method, ensuring it is a robust and reliable approach for automatic teacher identification on Pinterest.

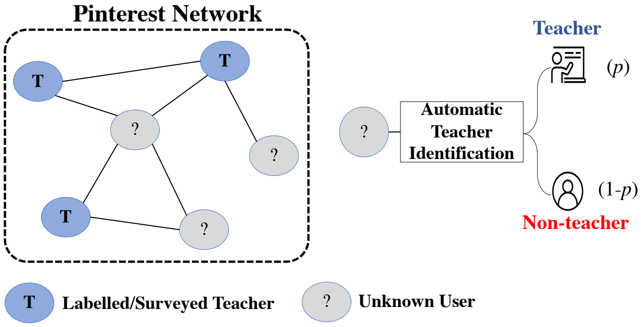


Figure 2: An overall illustration of our previously developed automatic teacher identification method

Table 1: Basic statistics of our constructed dataset

#Users (teachers)	**13,267**
#Pins	**1,162,983**
#Boards	**865,655**
#Followees	**11,84,940**
#Followers	**1,046,729**

Table 1 presents basic statistics of our compiled dataset. As the table indicates, our dataset comprises 13,267 teachers, who were either surveyed directly or identified automatically using our method. Furthermore, these teachers have curated over one million pins.

3.3 Diffusion Trees

We constructed diffusion trees for these resources to investigate the diffusion of curated resources on Pinterest. A diffusion tree is a directed graph, symbolized as $T = (U, E, p, r)$, representing the cascade of user information. Here, U denotes the set of users engaged in the diffusion, E is the set of directed edges between users in U, p is the pin being disseminated among users in U, and r is the origin or root of the tree—a teacher who initially curated the pin p. Each edge $e = (u_i, u_j) \in E$ suggests that the user $u_i \in U$ has received pin p from user $u_j \in U$ and subsequently re-pinned it. For instance, in Figure 1, user u_1 (the root) has curated a resource that has been disseminated throughout the network and re-pinned by users u_2, u_3, u_4, u_5, and u_6.

We constructed diffusion trees for 1,162,983 unique pins that our identified teachers curated, meaning the root node of each tree was one of the teachers we identified, as described in Section 3.1. It is important to note that not all users in U were necessarily teachers. Furthermore, we created trees for all types of resources curated by teachers, both educational and non-educational. This was done for two main reasons. Firstly, including non-educational pins allows us to better contextualize the diffusion patterns of educational resources compared to non-educational ones. Secondly, apart from studying the diffusion of teacher-curated resources on Pinterest, a secondary objective of this paper is to analyze teachers' general behavior on the platform. Therefore, examining the diffusion of all types of teacher-curated resources contributes to this secondary objective. Lastly, it is worth mentioning that our dataset of diffusion trees represents the largest dataset of diffused resources on Pinterest to date, offering the potential for future research on information diffusion on social media.

4. DIFFUSION MEASURES

We present three measures to characterize the diffusion process, inspired by those introduced in [22]. These measures aim to evaluate the large-scale and fast diffusion of educational resources on social media, as documented in previous studies [12, 20, 23]. Specifically, these measures are designed to echo the two critical aspects emphasized in prior research on the diffusion of educational resources on social media, particularly Pinterest: a) educational resources are disseminated on a large scale among teachers, and b) this dissemination of educational resources occurs rapidly [20, 12].

4.1 Volume

The first measure, *volume* (VL), is defined as the total number of nodes in a diffusion tree:

$$VL(T) = |U| \tag{1}$$

For instance, the volume of the tree depicted in Figure 1 is 6. Despite its apparent simplicity, the volume measure carries significant implications as it indicates how much information has diffused. Specifically, the count of users that have received the information is used in predicting or assessing the popularity of information on social media [24, 25]. Relevant to our study, we can determine the level of interest other users or teachers have in a teacher-curated resource by examining its volume.

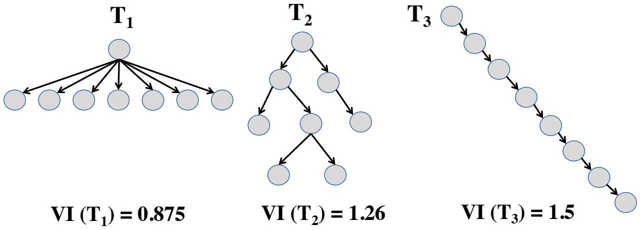
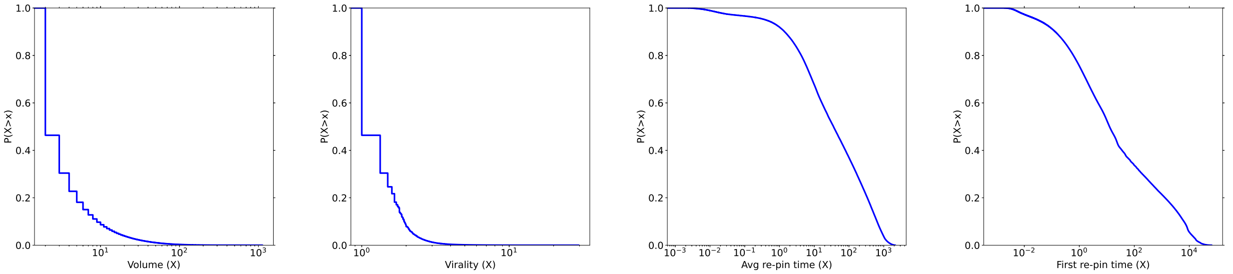


Figure 3: Three diffusion trees with the same volume but different virality values

4.2 Virality

While the volume measure is important, it only reports the number of individuals who have re-shared a resource. However, depending on the structure of a diffusion tree, the dissemination can take different forms. To illustrate this, Figure 3 presents three distinct diffusion trees, all having a volume of 8 but exhibiting very different forms of dissemination. In T_1, there is a broadcast from the root to other nodes, with only the root participating in the information propagation. In contrast, T_2 involves more nodes in the diffusion process. T_3 represents an extreme scenario with a chain-wise 'deep' tree, where the message has been passed on consecutively. Distinguishing between diffusion scenarios based on their tree structure provides insight into the virality and penetration of a message across the network [22].



(a) Volume (Eq. 1) (b) Virality (Eq. 2) (c) Average re-pin time (Eq. 3) (d) The first re-pin time (Eq. 4)

Figure 4: The CCDF plots of the defined diffusion measures (x-axes are in log scale)

Table 2: Some statistics of the introduced diffusion measures of the constructed diffusion trees

Diffusion Measure	Min	Max	Mean	Median	Std	top 0.1%	top 0.01%
Volume	2	1,129	5.4	2	13.58	> 174	> 434
Virality	1	29.72	1.33	1	0.54	> 5.99	> 11.45
ART	0.0012	2,159.4	192.4	35.8	317.3	> 1,950.7	> 2,113.2
FRT	0.0008	65,655.0	1,814.4	12.5	4,975.0	> 45,020.7	> 56,960.9

Therefore, we define the *virality* (VI) of a diffusion tree as follows:

$$VI(T) = \frac{2}{(|U|) \times (|U|-1)} \sum_{\forall u_i, u_j \in U} d(u_i, u_j) \qquad (2)$$

Here, $d(u_i, u_j)$ represents the shortest distance between two users u_i and u_j in the diffusion tree T. The sum of the shortest distances between nodes in a graph is known as the Wiener Index [26, 27]. The term $\frac{2}{(|U|)\times(|U|-1)}$ normalizes the Wiener Index. Based on this measure, we can observe that T_3 has the highest virality among the trees in Figure 3.

4.3 Velocity

Alongside volume and virality, the speed of diffusion is also crucial. Previous studies have highlighted the rapid diffusion of educational resources on social media, especially Pinterest, making these platforms highly appealing to teachers [28, 23]. Thus, our third diffusion measure pertains to the velocity (or speed) of diffusion. For this, we introduce two metrics.

The first metric is the *average re-pin time*, which calculates the average time between two re-pins in the diffusion tree. The average re-pin time (ART) for a diffusion tree is defined as:

$$ART(T) = \frac{1}{|U|-1} \sum_{\forall e \in T} u_j(t) - u_i(t) \qquad (3)$$

Here, (u_i, u_j) is an edge in the diffusion tree and $u_i(t)$ ($u_j(t)$) represents the re-pin time by user u_i (u_j). In Eq. 3, we have subtracted $u_i(t)$ from $u_j(t)$ as the user u_i received the pin earlier. Given the rapid diffusion of information on social media, we use an hour as the time scale. The ART for the example tree shown in Figure 1 is 46.2 hours.

However, sometimes a resource can continue to be diffused for an extended period (for example, months), which can result in a large ART. Therefore, to better capture the diffusion velocity, we define the *first re-pin time* (FRT). It represents the time duration from the initial curation of a pin to its first re-pin:

$$FRT(T) = min\{u_i(t) - r(t)\} \;\; s.t. \;\; (r, u_i) \in E \qquad (4)$$

Here, $r(t)$ denotes the time the root curated the pin. The FRT for the example tree in Figure 1 is 5.16 hours.

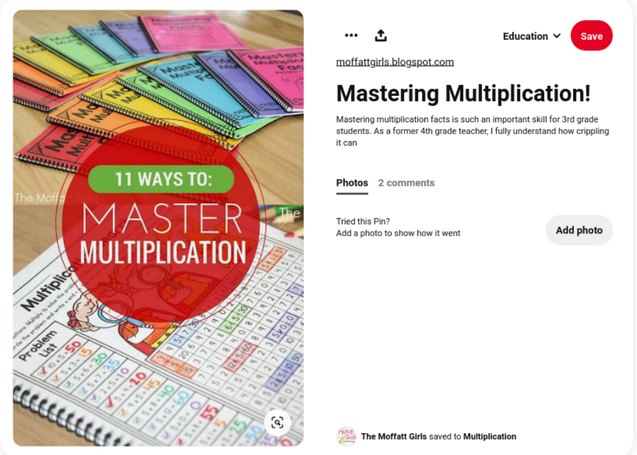
5. DIFFUSION ANALYSIS

In this section, we examine the diffusion trees we have constructed. First, in Section 5.1, we provide statistical data on diffusion measures. Next, in Section 5.2, we discuss how different resource types are diffused. Lastly, in Section 5.3, we explore the relationship between the earlier diffusion measures and specific attributes related to teachers.

5.1 Statistics of Diffusion Measures

This section delves into the statistics and distributions of the three diffusion measures. Table 2 provides specific statistics about virality, volume, and velocity measures. In addition, the CCDF (complementary cumulative distribution function) of the diffusion measures is depicted in Figure 4.

As depicted in Figures 4a and 4b, both volume and virality exhibit a power-law distribution. This suggests that while most resources have low volume and virality, a small percentage displays exceptionally high values for these measures. Further, as per Table 2, the top 0.1% of diffused resources exhibit a volume and virality exceeding 174 and 5.99 hours,



moffattgirls.blogspot.com

Mastering Multiplication!

Mastering multiplication facts is such an important skill for 3rd grade students. As a former 4th grade teacher, I fully understand how crippling it can

Photos 2 comments

Tried this Pin?
Add a photo to show how it went

The Moffatt Girls saved to Multiplication

Figure 5: A sample of a popular pin from *moffattgirls.blogspot.com* account, a prolific educator, that has been received (re-pinned) by 936 other users

respectively. This indicates that certain resources curated by teachers have gained considerable popularity. These findings align with prior studies on the virality and popularity of information on social media, demonstrating that some information can become significantly viral across the network [29, 30, 22]. On average, approximately five users have re-pinned each teacher-curated resource on Pinterest.

Contrary to volume and virality, the velocity measures do not adhere to a power-law distribution as seen in Figures 4c and 4d. Furthermore, a notable disparity exists between the mean and median for ART and FRT. While the median average re-pin and first re-pin times are relatively short (35.56 and 12.56 hours, respectively), their means are skewed due to the presence of outliers.

In conclusion, teacher-curated resources diffuse rapidly and reach a significant number of other users on Pinterest, including other teachers. Although this observation has been suggested in anecdotal reports [20, 12], our study offers the first large-scale data-driven analysis to confirm it.

5.2 Resource Attributes and Diffusion

In this part, we explore the diffusion measures in relation to two key attributes of pins: their topics and domains.

5.2.1 *Topic*

Every pin on Pinterest is associated with a pre-defined topic (or category), such as *travel*, *education*, or *fashion*. Figure 6a presents the average volume value for each topic. As evidenced by this figure, pins categorized under *education* exhibit the highest volume, with each such pin being received by an average of six users on Pinterest. Interestingly, *kids* ranks second in terms of volume, a fact that could be partially attributed to this topic's similarity to *education* and its appeal to teachers, particularly for resources specific to pre-kindergarten or homeschooling. All other topics exhibit lower volumes, generally below 4. Given that the dominant topic is *education*, and there is limited data for other topics, these topics demonstrate relatively high standard deviations.

Figure 6b displays the average virality value for each topic. As with volume, *education* also records the highest virality, indicating the extensive penetration of teacher-curated educational resources across Pinterest. The topic *kids* also showcases a relatively high average virality value. Moreover, comparing the volume and virality values in Figure 6a and Figure 6b suggests that high volume does not necessarily equate to high virality. For example, pins categorized under *quotes* exhibit relatively high virality, but their volume is not as impressive.

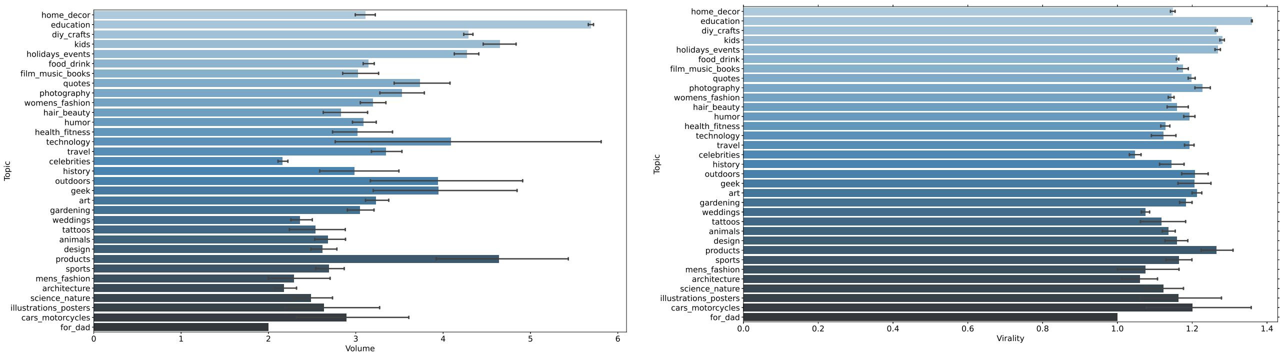
Figure 7 depicts the median values for the average and first re-pin times. We opted to use the median for these plots since, as discussed in Section 5.1, the ART and FRT values of our constructed diffusion trees include some outliers. Moreover, specific topics have limited data, resulting in skewed velocity measures. Therefore, for clarity, we present the median velocity measures for topics with a pin proportion of at least 10%.

From the results of the velocity measures, we observe two key points. Firstly, the topic *education* has both a short average re-pin time and first re-pin time. Specifically, the median of the first re-pin time for *education* is just 12 hours, suggesting that a teacher-curated resource takes roughly half a day to be received by another user on Pinterest. This underscores the rapid diffusion of educational materials across Pinterest. Secondly, the average re-pin time is generally longer than the first re-pin time. We posit that this may be due to a user quickly saving a pin curated by the root, with the pin then spreading across the network at a slower rate. However, there are a few exceptions to this, such as *travel* and *art*. This could be attributed to the unique appeal of these topics to teachers, whose pins may take some time to attract attention initially. However, once they gain traction, they diffuse more rapidly.

5.2.2 *Domain*

Pinterest allows users to pin resources from anywhere on the web. Given this property, examining the diffusion of pins from various sources becomes essential. In this section, we analyze the diffusion of teacher-curated resources based on the domains of their sources. For this analysis, we consider only the top 10 domains preferred by teachers. Figure 8 presents the average volume and virality values for these top 10 domains. Figure 9 displays the median of the average re-pin time and the first re-pin time for the exact top 10 domains. From these results, we draw the following observations:

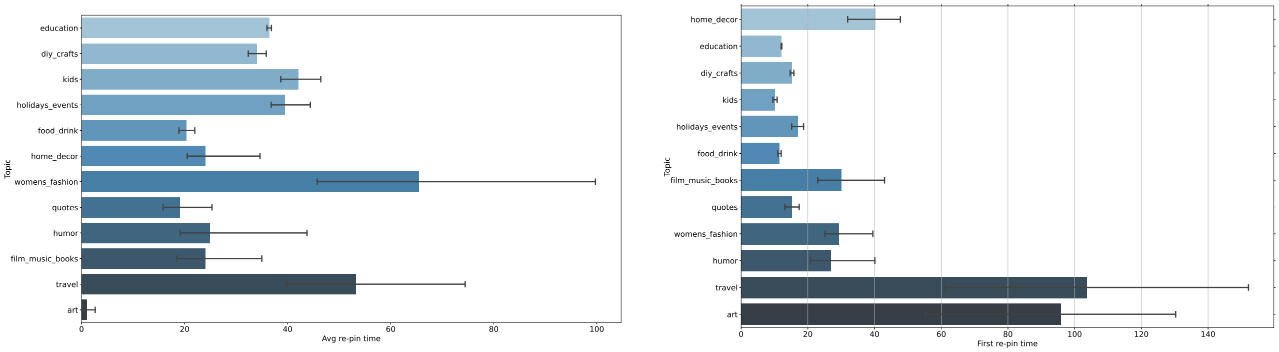
- ❏ Except for *youtube.com* and *Uploaded by User* (resources directly uploaded from the user's device), the remaining domains are predominantly education-related, for example, *moffattgirls.blogspot.com*. Interestingly, pins from *moffattgirls.blogspot.com* record the highest volume. This blog is managed by a former elementary school teacher who exclusively creates educational materials. Further investigation reveals that this teacher is highly active and influential on *teacherspayteachers.com*– the largest online marketplace for instructional resources. Therefore, it is not surprising that her educational materials garner significant interest.



(a) Volume

(b) Virality

Figure 6: Mean of volume and virality per topic



(a) Average re-pin time

(b) The first re-pin time

Figure 7: Median of the velocity measures for the top topics

Additionally, the pins from this domain exhibit high virality and rapid diffusion. Such active and inspiring teachers exemplify the influential role of teachers on social media and the impact they can have on their peers in the digital age. Figure 5 showcases a popular sample pin from *moffattgirls.blogspot.com*.

- Materials from *teacherspayteachers.com* also demonstrate high volume and virality, suggesting the popularity of educational materials from this source. Interestingly, the velocity measures for pins from *teacherspayteachers.com* reveal a short first re-pin time but a relatively longer average re-pin time. This is because pins from this popular source continue to be diffused across Pinterest over an extended period, resulting in a longer average re-pin time.

- Another observation is the long first re-pin time for pins from the *Uploaded by User* domain. We surmise that this may be due to the following reason: since these pins do not originate from a specific internet website (i.e., they have no domain), other users might be hesitant to save them promptly, possibly due to trust issues. However, once these pins gain (intial) popularity, they become more widespread and diffuse across the network.

5.3 Teacher Attributes and Diffusion

In addition to the resource itself, a resource producer (e.g., a teacher) can also influence the diffusion process [31]. There exists a substantial body of literature focused on identifying influential spreaders in social media, based on their attributes [32, 33, 34]. Consequently, in this section, we investigate whether teacher attributes are associated with the diffusion of the resources they curate. To achieve this, we considered the following ten teacher-related attributes and analyzed their relationship with diffusion metrics:

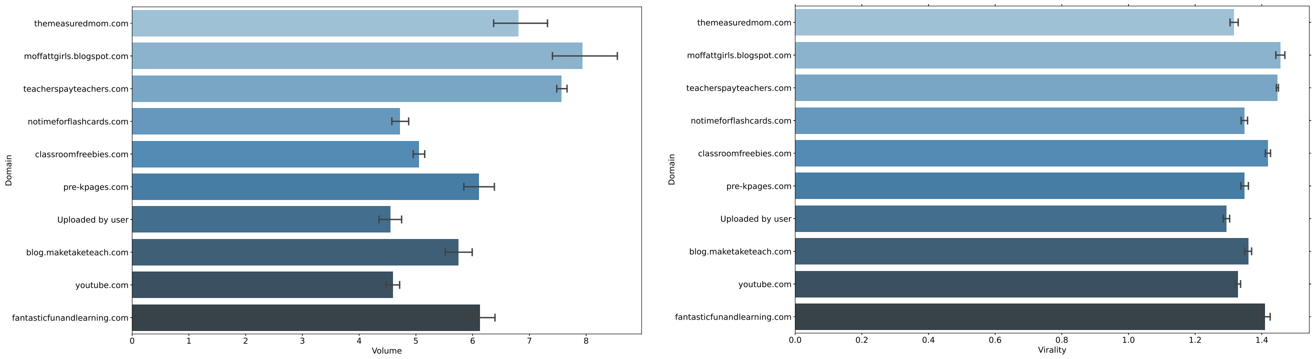
1. Number of pins: Assessing whether a teacher's activity level leads to widespread and fast diffusion of their materials.

2. Number of boards: Similar to the number of pins, this attribute also evaluates the impact of a teacher's activity level on diffusion.

3. Number of followers: Investigating whether resources of a teacher with more followers have a higher chance of being disseminated in the network.

4. Number of followees: Examining how this attribute affects the diffusion measures.

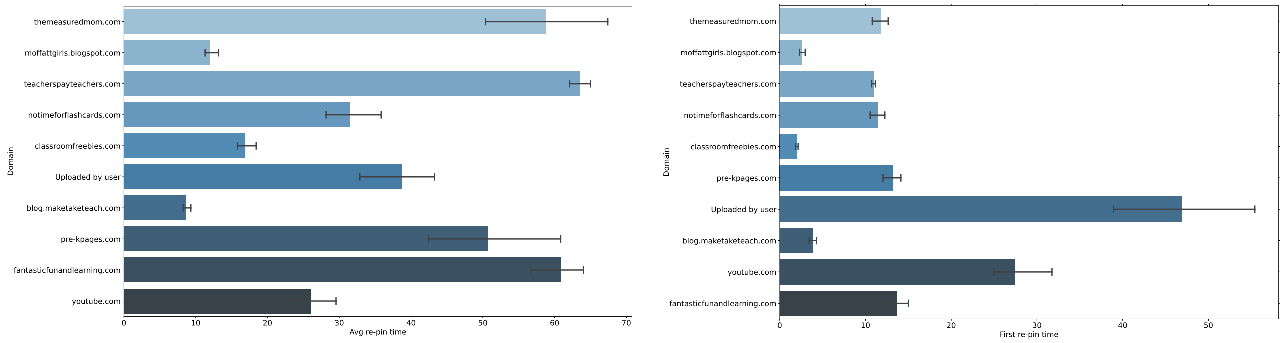
5. Total number of friends (followers and followees combined): Analyzing the impact of this attribute on diffusion measures.



(a) Volume

(b) Virality

Figure 8: Mean of volume and virality for the top 10 domains of teacher-curated resources



(a) Average re-pin time

(b) The first re-pin time

Figure 9: Median of the velocity measures for the top 10 domains of teacher-curated resources

6. Reciprocity: Investigating the relationship between reciprocity and diffusion, in order to determine whether having a stronger connection between a teacher and their online friends affects the diffusion.

7. Eigenvector centrality: Assessing whether resources of more central teachers have a higher chance to be adopted by other users and perhaps at a faster rate.

8. Betweenness centrality: Similar to eigenvector centrality, this attribute also evaluates the structural importance of a teacher in the network.

9. Closeness centrality: Another measure of centrality, investigating the influence of a teacher's structural importance on the diffusion of resources.

10. Local clustering coefficient: Quantifying how close a user's neighbors are to a complete graph (a clique), as previous studies [35, 36] have shown that cliques in school-level teacher networks can lead to better diffusion of information.

To explore the relationship between teacher attributes and diffusion measures, we conducted four regression analyses. In each analysis, teacher attributes served as the independent variables, while the corresponding diffusion measure acted as the dependent variable. Our goal was to determine the extent to which each teacher's attributes could explain a diffusion measure. It is important to note that we focused only on teachers who were the roots of the diffusion trees, as our aim was to identify the attributes of the pin producers, not those who further engaged in re-pinning. Consequently, a teacher could be associated with multiple diffusion trees as the root. In order to perform a teacher-level analysis, we aggregated the values of each diffusion measure for all diffusion trees associated with a teacher. For volume and virality, we calculated the mean values. For the velocity measures, we opted for the median, as it provides a more accurate estimation than the mean, as previously discussed. Finally, we utilized the statsmodels package [37] in Python to fit ordinary least squares (OLS) for each regression analysis. The results are shown in Tables 3 and 4. We make the following observations based on these results:

❏ The adjusted R-squared values are high for volume and virality, indicating that teacher attributes can significantly explain these measures. Conversely, these attributes fail to sufficiently explain the velocity measures, as reflected by the low adjusted R-squared values. This interpretation is corroborated by examining the mean squared errors: while these values are low for volume and virality, suggesting a good model fit, they are high for the velocity measures, indicating a less accurate fit.

❏ Regarding the number of pins, the coefficients for all

Table 3: Regression analysis results of predicting volume and virality using teacher attributes

	Volume				Virality			
Attribute	Coefficient	Std error	t	P > |t|	Coefficient	Std error	t	P > |t|
#Pins	4.449e-07	2.44e-06	0.182	0.855	-2.419e-06	2.29e-07	-10.574	0.000
#Boards	-0.0011	0.000	-3.049	0.002	-0.0001	3.31e-05	-3.861	0.000
#Followers	-0.0005	0.000	-3.590	0.000	-1.318e-05	1.41e-05	-0.936	0.349
#Followees	0.0003	0.000	2.539	0.011	-3.879e-05	9.63e-06	-4.029	0.000
#Friends	-0.0003	9.35e-05	-2.976	0.003	-5.197e-05	8.77e-06	-5.923	0.000
Reciprocity	0.2359	0.084	2.816	0.005	0.0894	0.008	11.379	0.000
Eigenvector Cent	54.8268	12.913	4.246	0.000	2.5898	1.211	2.138	0.033
Betweenness Cent	28.2263	75.308	0.375	0.708	24.0530	7.065	3.405	0.001
Closeness Cent	7.4408	0.190	39.110	0.000	3.5286	0.018	197.698	0.000
LCC	1.7113	0.283	6.039	0.000	0.6131	0.027	23.064	0.000
	Mean squared error: 2.19 Adj. R-squared: 0.539				Mean squared error: 0.04 Adj. R-squared: 0.965			

Table 4: Regression analysis results of predicting velocity using teacher attributes

	Average re-pin time				The first re-pin time			
Attribute	Coefficient	Std error	t	P > |t|	Coefficient	Std error	t	P > |t|
#Pins	-0.0015	0.000	-4.077	0.000	0.0308	0.005	-6.450	0.000
#Boards	0.1079	0.061	1.781	0.075	1.3398	0.691	1.940	0.052
#Followers	0.0096	0.022	0.440	0.660	0.3589	0.294	1.223	0.222
#Followees	0.0020	0.016	0.127	0.899	-0.3620	0.201	-1.804	0.071
#Friends	0.0116	0.014	0.822	0.411	-0.0031	0.183	-0.017	0.986
Reciprocity	-143.80	17.34	-8.290	0.000	-1005.0319	163.868	-6.133	0.000
Eigenvector Cent	-8090.16	1884.5	-4.293	0.000	-8.951e+04	2.53e+04	-3.544	0.000
Betweenness Cent	8936.45	1.1e+04	0.811	0.417	1.404e+05	1.47e+05	0.953	0.340
Closeness Cent	607.29	36.66	16.5	0.000	85375.8527	372.131	14.446	0.000
LCC	410.17	62.89	6.522	0.000	4871.5493	554.231	8.790	0.000
	Mean squared error: 85667.98 Adj. R-squared: 0.263				Mean squared error: 18323220.30 Adj. R-squared: 0.143			

measures were generally low. This suggests that a high activity rate of a pin's producer does not necessarily correlate with the diffusion of pins. This is logical, as simply saving more pins and creating more boards on Pinterest does not guarantee widespread diffusion of these resources. The only notable exception was the number of boards for the first re-pin time, where the coefficient was both positive and relatively large. This can be attributed to the fact that Pinterest users can follow a board independently, without having to follow its curator. Consequently, the more boards a user has, the higher the likelihood that someone could quickly re-pin from any of these boards. However, based on our findings, such rapid adoption does not necessarily translate to high volume and virality for the pin.

❑ The coefficients related to the number of connections, namely the number of followers, followees, and friends, were notably low. Although the coefficient of the number of followers was relatively high for the first re-pin time, it lacked statistical significance, as indicated in the P>|t| column. We posit that the low coefficient values for the number of connections can be attributed to Pinterest's nature as a social curation platform. On this website, users have the ability to re-pin resources from others without necessarily following them.

❑ Reciprocity exhibits a low coefficient for virality, yet a relatively high one for volume. Teachers with high reciprocity tend to have strong relationships with their online friends, fostering a trusting environment for repinning their resources. However, virality is a complex measure that cannot be sufficiently explained by reciprocity alone. As for the velocity measures, the coefficients of reciprocity are large and negative, a phenomenon that warrants further exploration.

❑ The most significant observation from this section is the relationship between the centrality metrics and the volume and virality. With the exception of betweenness centrality for volume, the centrality metrics provide a comprehensive explanation for both virality and volume. This is likely due to the centrality metrics taking into account the network's structure, a crucial factor in information diffusion. For example, a teacher with high eigenvector centrality is connected to other users with high centrality. Consequently, when these central neighbors re-pin a resource, the likelihood of wide diffusion increases due to their significant structural influence. Furthermore, both closeness and betweenness centrality are related to the shortest paths in the network, which play a critical role in the diffusion of information [38].

❑ As previously noted, the local clustering coefficient plays a pivotal role in the diffusion of information within school-level teacher networks [35, 36]. Moreover, findings from this section of the dissertation demonstrate that this attribute is equally significant in the diffusion of information across the network of teachers on Pinterest.

From the observations discussed, we can infer that teacher attributes have a substantial impact on the volume and virality of the resources they curate. Specifically, a teacher's structural characteristics at the network level play a crucial role in determining their resources' volume and virality. However, these attributes do not adequately predict the speed at which these resources are diffused.

6. CONCLUSION AND FUTURE WORK

This paper extensively analyzed the diffusion process of teacher-curated resources on Pinterest. Our first task involved constructing a comprehensive set of diffusion trees for these resources on the platform. Subsequently, we defined three critical measures to capture the essence of the diffusion process: volume, virality, and velocity. Finally, our in-depth analysis revealed that educational materials experience wide and rapid dissemination across Pinterest.

To further our understanding of diffusion dynamics, we executed multiple regression analyses to identify the teacher attributes that significantly influence the diffusion process. Our findings underscored the crucial role of structural attributes in the diffusion of teacher-curated resources on Pinterest. This is an important insight, demonstrating the relevance of network-level structural characteristics in predicting the volume and virality of resources. However, our study also indicated that these teacher attributes do not adequately explain the speed of diffusion, pointing to the complexity of the diffusion process and suggesting the need for further investigation.

Our large-scale, data-driven study not only deepens the understanding of how teacher-curated materials diffuse on Pinterest but also sets the stage for future research. The insights garnered here could be instrumental in optimizing information dissemination strategies on social curation platforms like Pinterest and beyond. By identifying the key factors that influence diffusion, educational stakeholders can harness these attributes to enhance the reach and impact of curated resources. Additionally, our findings may inspire researchers to delve deeper into the mechanisms underlying the diffusion process, encouraging the exploration of other factors and attributes we have not covered in this study. This research opens up a rich avenue for further inquiry and innovation in information diffusion in online educational networks.

There are a couple of interesting future directions:

➜ **In-depth Analysis of Velocity Measures:** Our study indicated that teacher attributes do not sufficiently explain the speed of diffusion (velocity). Therefore, future research could focus on a more detailed investigation into the factors influencing the velocity of resource diffusion. This might include considering additional user attributes, the nature of the content being shared, or even network-level factors such as pins' timing and user engagement dynamics over time.

➜ **Role of Content Characteristics:** This study primarily focused on the role of teacher attributes in the diffusion process. Future research could extend this to consider the curated resources' characteristics. This could involve analyzing the resources' content, format, topic, or even aesthetic appeal and how these factors might influence their diffusion.

➜ **Temporal Analysis of Diffusion:** Another interesting direction could be the temporal analysis of diffusion processes. How do teacher attributes and the diffusion of their resources evolve over time? Longitudinal studies could provide further insights into the dynamic nature of information diffusion on Pinterest.

➜ **Cross-platform Studies:** This research was focused on Pinterest. Future studies could examine diffusion processes on other social curation platforms or across multiple platforms. Such studies could reveal platform-specific characteristics influencing diffusion and offer a comparative perspective.

➜ **Impact of Algorithmic Recommendations:** Pinterest, like many other platforms, uses recommendation algorithms to suggest pins to users. Future research could explore how these algorithms influence the diffusion process. This could involve studying the interaction between user behavior and the platform's algorithmic curation in shaping diffusion patterns.

7. REFERENCES

[1] Hamid Karimi, Tyler Derr, Kaitlin T Torphy, Kenneth A Frank, and Jiliang Tang. A roadmap for incorporating online social media in educational research. *Teachers College Record*, 121(14):1–24, 2019.

[2] Hamid Karimi, Tyler Derr, Kaitlin T Torphy, Kenneth A Frank, and Jiliang Tang. Towards improving sample representativeness of teachers on online social media: A case study on pinterest. In *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II 21*, pages 130–134. Springer International Publishing, 2020.

[3] Kaitlin Torphy Knake, Hamid Karimi, Sihua Hu, Kenneth A Frank, and Jiliang Tang. Educational research in the twenty-first century: Leveraging big data to explore teachers' professional behavior and educational resources accessed within pinterest. *The Elementary School Journal*, 122(1):86–111, 2021.

[4] Hamid Karimi. *Teachers in social media: a data science perspective*. PhD thesis, Michigan State University, 2021.

[5] Hamid Karimi, Jiliang Tang, Xochitl Weiss, and Jiangtao Huang. Automatic identification of teachers in social media using positive unlabeled learning. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 643–652. IEEE, 2021.

[6] Hamid Karimi, Kaitlin Torphy Knake, and Kenneth A Frank. Teachers in social media: A gender-aware behavior analysis. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 1842–1849. IEEE, 2022.

[7] Kaitlin Torphy Knake, Zixi Chen, Xiuqi Yang, and Jordan Tait. Pinterest curation and student achievement: The effects of elementary mathematics resources on students' learning over time. *the elementary school journal*, 122(1):57–85, 2021.

[8] Kaitlin Torphy, Sihua Hu, Yuqing Liu, and Zixi Chen. Teachers turning to teachers: teacherpreneurial behaviors in social media. *American Journal of Education*, 127(1):49–76, 2020.

[9] Christine Greenhow, Sarah M Galvin, Diana L Brandon, and Emilia Askari. A decade of research on k-12 teaching and teacher learning with social media: Insights on the state of the field. *Teachers College Record*, 122(6):n6, 2020.

[10] V. Darleen Opfer, Julia H. Kaufman, and Lindsey E. Thompson. *Implementation of K-12 state standards for mathematics and English Language Arts and literacy: Findings from the American Teacher Panel.* RAND Corporation, Santa Monica, CA, 2016.

[11] Statista. Number of monthly active pinterest users worldwide from 2nd quarter 2016 to 3rd quarter 2021 (in millions)., 2021.

[12] Sihua Hu, Kaitlin T Torphy, Amanda Opperman, Kimberly Jansen, and Yun-Jia Lo. What do teachers share within socialized knowledge communities: A case of pinterest. *Journal of Professional Capital and Community*, 2018.

[13] Jeffrey Carpenter, Amanda Cassaday, and Stefania Monti. Exploring how and why educators use pinterest. In *Society for Information Technology & Teacher Education International Conference*, pages 2222–2229. Association for the Advancement of Computing in Education (AACE), 2018.

[14] Eric Gilbert, Saeideh Bakhshi, Shuo Chang, and Loren Terveen. " i need to try this"? a statistical overview of pinterest. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2427–2436, 2013.

[15] Jinyoung Han, Daejin Choi, Byung-Gon Chun, Ted Kwon, Hyun-chul Kim, and Yanghee Choi. Collecting, organizing, and sharing pins in pinterest: interest-driven or social-driven? *ACM SIGMETRICS Performance Evaluation Review*, 42(1):15–27, 2014.

[16] Daehoon Kim, Jae-Gil Lee, and Byung Suk Lee. Topical influence modeling via topic-level interests and interactions on social curation services. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 13–24. IEEE, 2016.

[17] Regina Kasakowskij, Thomas Kasakowskij, and Kaja Fietkiewicz. Pinterest: A unicorn among social media? an investigation of the platform's quality and specifications. In *ECSM 2020 8th European Conference on Social Media*, page 399. Academic Conferences and publishing limited, 2020.

[18] Stephanie Schroeder, Rachelle Curcio, and Lisa Lundgren. Expanding the learning network: How teachers use pinterest. *Journal of Research on Technology in Education*, 51(2):166–186, 2019.

[19] Sihua Hu, Kaitlin T Torphy, Kim Evert, and John L Lane. From cloud to classroom: Mathematics teachers' planning and enactment of resources accessed within virtual spaces. *Teachers College Record*, 122(6):n6, 2020.

[20] Yuqing Liu, Kaitlin T Torphy, Sihua Hu, Jiliang Tang, and Zixi Chen. Examining the virtual diffusion of educational resources across teachers' social networks over time. *Teachers College Record*, 122(6):n6, 2020.

[21] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

[22] Jinyoung Han, Daejin Choi, Jungseock Joo, and Chen-Nee Chuah. Predicting popular and viral image cascades in pinterest. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, 2017.

[23] Christine Greenhow, Sarah M Galvin, and K Bret Staudt Willet. What should be the role of social media in education? *Policy Insights from the Behavioral and Brain Sciences*, 6(2):178–185, 2019.

[24] Thi Bich Ngoc Hoang and Josiane Mothe. Predicting information diffusion on twitter–analysis of predictive features. *Journal of computational science*, 28:257–264, 2018.

[25] Bo Wu and Haiying Shen. Analyzing and predicting news popularity on twitter. *International Journal of Information Management*, 35(6):702–711, 2015.

[26] Andrey A Dobrynin, Roger Entringer, and Ivan Gutman. Wiener index of trees: theory and applications. *Acta Applicandae Mathematica*, 66(3):211–249, 2001.

[27] Harry Wiener. Structural determination of paraffin boiling points. *Journal of the American chemical society*, 69(1):17–20, 1947.

[28] Kaitlin T Torphy, Diana L Brandon, Alan J Daly, Kenneth A Frank, Christine Greenhow, S Hua, and Martin Rehm. Social media, education, and digital democratization. *Teachers College Record*, 122(6):1–7, 2020.

[29] Lilian Weng, Filippo Menczer, and Yong-Yeol Ahn. Virality prediction and community structure in social networks. *Scientific reports*, 3(1):1–6, 2013.

[30] Piia Varis and Jan Blommaert. Conviviality and collectives on social media: Virality, memes, and new social structures. *Multilingual Margins: A journal of multilingualism from the periphery*, 2(1):31–31, 2015.

[31] Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A Zighed. Information diffusion in online social networks: A survey. *ACM Sigmod Record*, 42(2):17–28, 2013.

[32] Qian Li, Tao Zhou, Linyuan Lü, and Duanbing Chen. Identifying influential spreaders by weighted leaderrank. *Physica A: Statistical Mechanics and its Applications*, 404:47–55, 2014.

[33] Ling-ling Ma, Chuang Ma, Hai-Feng Zhang, and Bing-Hong Wang. Identifying influential spreaders in complex networks based on gravity formula. *Physica A: Statistical Mechanics and its Applications*, 451:205–212, 2016.

[34] Frank Bauer and Joseph T Lizier. Identifying influential spreaders and efficiently estimating infection numbers in epidemic models: A walk counting approach. *EPL (Europhysics Letters)*, 99(6):68007, 2012.

[35] Kenneth Frank, Yun-jia Lo, Kaitlin Torphy, and Jihyun Kim. Social networks and educational opportunity. In *Handbook of the Sociology of Education in the 21st Century*, pages 297–316. Springer, 2018.

[36] William Penuel, Kenneth Frank, Min Sun, Chong

Kim, and Corinne Singleton. The organization as a filter of institutional diffusion. *Teachers college record*, 115(1):1–33, 2013.

[37] Skipper Seabold and Josef Perktold. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.

[38] Anastasia Mochalova and Alexandros Nanopoulos. On the role of centrality in information diffusion in social networks. 2013.

Semantic Topic Chains for Modeling Temporality of Themes in Online Student Discussion Forums

Harshita Chopra
Adobe Research, India
harshitac@adobe.com

Yiwen Lin
University of California, Irvine
yiwenl21@uci.edu

Mohammad Amin Samadi
University of California, Irvine
masamadi@uci.edu

Jacqueline G. Cavazos
University of California, Irvine
jacqueline.cavazos@uci.edu

Renzhe Yu
Columbia University
renzheyu@tc.columbia.edu

Spencer Jaquay
University of California, Irvine
sjaquay@uci.edu

Nia Nixon
University of California, Irvine
dowelln@uci.edu

ABSTRACT

Exploring students' discourse in academic settings over time can provide valuable insight into the evolution of learner engagement and participation in online learning. In this study, we propose an analytical framework to capture topics and the temporal progression of learner discourse. We employed a Contextualized Topic Modeling technique on messages posted by undergraduates in online discussion forums from Fall 2019 to Spring 2020. We further evaluated if topics were originating from specific courses or more generally distributed across multiple courses. Our results suggested a significant increase in the number of general topics after the onset of the pandemic, suggesting emergent topics being discussed in a range of courses. In addition, using Word Mover's Distance, we examined the semantic similarity of topics in adjacent months and constructed topic chains. Our findings indicated that previously course-centric topics such as public health developed into more general discussions that emphasize inequities and healthcare during the pandemic. Furthermore, emergent topics around students' lived experiences underscored the role of discussion forums in capturing educational experiences temporally. Finally, we discuss the implications of current findings for post-pandemic higher education and the effectiveness of our framework in exploring unstructured large-scale educational data.

Keywords

Topic Modeling, Topic Detection and Tracking, Natural Language Processing, Discourse Analysis

1. INTRODUCTION

Social learning theories suggest that learning occurs through interaction with peers and instructors[52]. In fully online learning contexts, interpersonal interaction plays an even more important role and discussion forum is one of the most commonly used tool to enable interactions and facilitate classroom community. As such, examining text and language in a discussion forum allows us to gain insights on learning since language plays an instrumental role in promoting thinking and knowledge construction as well as social activities occurring in computer-mediated environments[51]. While the use of discussion forums has been prominent on informal learning platforms such as Massive Open Online Courses (MOOCs) [4], the adoption of discussion forums for social learning and communication across accredited institutions has been less systematic until the contingency shift to remote learning due to COVID-19.

In late 2019, news began to spread about a respiratory illness appearing throughout parts of the globe. By March 2020, the World Health Organization officially declared COVID-19 a global pandemic [57]. In response to the global health crisis, universities and schools quickly suspended in-person classes and shifted to fully online instruction[50]. Although online learning and teaching are already prevalent, the urgent shift to online education created a nontrivial disruption to students' educational experience[50]. During this unprecedented time, undergraduate students were confronted with new health concerns, challenges adapting to online learning, and seeking ways to build the relationships and connect with peers that are essential for students' college success. The outbreak has also posed challenges for instructors to ensure instructional delivery and quality through a variety of tools such as video conferencing, forum discussions and assignment submission through Learning Management Systems (LMS).

As learners were prompted to fully remote learning, the amount of educational data generated day by day became unprecedented. Since the pandemic was a major disruption to regular instruction, the shift to fully remote instruction derived a need to harvest insight into how learning activities were impacted [33]. Recent advances in data mining tech-

H. Chopra, Y. Lin, M. A. Samadi, J. G. Cavazos, R. Yu, S. Jaquay, and N. Nixon. Semantic topic chains for modeling temporality of themes in online student discussion forums. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 67–78, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115691

niques provide efficient and promising tools to effectively process large-scale educational data to model learner behavior in real-time [1]. Moreover, leveraging objective indicators rather than self-report data allows a non-intrusive way to explore the consequences of the transition to online learning and the impact of COVID-19 on the educational process [11]. Digital trace data generated from learner's activities in LMS are not only useful in tracking engagement patterns and predicting learning outcomes across informal and formal education context [30], but features such as textual data also allow for non-intrusive means to analyze social and cognitive dynamics compared to interviews or questionnaire for data sampling [46]. The applications of Natural Language Processing (NLP) techniques in education have been proven to be powerful and effective tools for modeling learner behavior and extracting meaningful information to enhance our understanding of social interactions in computer-mediated learning environments [20, 32, 17, 19, 22]. A recent systematic review also identifies textual analysis of asynchronous discussion forums arises as an emergent theme in learning analytics given the increasing use of online LMS platforms[2]. This suggests that increased attention in the field of learning analytics has been given to exploratory approaches to harvesting insights from educational discourse.

Given the COVID-19 circumstances and the constantly changing nature of the pandemic and policy responses, automated methods focused on detecting the temporal aspect of learning activities could be valuable in providing insights to policymakers and instructors on the overall changing landscape in online learning. Indeed, text mining techniques offer a viable way to extract meaningful information and unravel latent patterns from large-scale educational data. Taken together, we built on our previous study [15] to propose an analytical framework that characterizes the evolution of topics in unstructured student discourse present in online discussion forums. We applied this framework to derive a series of semantically meaningful topic chains that we believe reflect the changes in online learning activities due to the emergency shift to remote learning and the COVID-19 pandemic influence.

This paper employs novel NLP approaches to explore the emergent topics in LMS and reveal the temporal development of teaching and learning activities in online discussion forums. Specifically, we investigate the topics before and after universities transitioned to a fully remote format to reflect the organic responses from teachers and students to this significant disruption in teaching and learning. As a methodological contribution, our framework combines state-of-the-art topic modeling techniques (i.e., Contextualized Topic Modeling and Word Mover Distance) to construct semantically relevant topics on a month-to-month granularity. This framework offers an opportunity to examine topical evolution in educational data. Despite the popularity of topic modeling in social media research, it remains a niche cluster in educational research [2] and focuses on static content classification[25]. In contrast, the temporal dynamics of topics highlighted in our framework would be a unique contribution in this area. Additionally, we extracted course centrality as a course-level feature to further contextualize topic chains. This allows us to distinguish topics that are specific to a discipline from more generally occurring topics in the entire forum environment. Finally, our study presents an exploratory rather than prescriptive view based on emergent student discourse revealing the impact of larger societal events on educational activities and a transparent analytic process to harvest information that aids decision-making.

The rest of the paper is organized as follows. First, we provide a brief overview of text mining practices in education in recent years. Second, we describe specific topic modeling techniques for processing large-scale text corpus. We then introduce our research questions and the methodological approach of the current study. Finally, we present our findings with a discussion on the implications of our study and the potential adaptation of this analytical framework for the broader education data mining community.

2. BACKGROUND

2.1 Text Mining in Education

Language is a channel for expressing people's opinions and experiences. The advances in computational linguistics offer ways to systematically explore these experiences and capture trends in discourse through individuals' language use. In educational contexts, the exponential growth of learner data generated in the digital environment has prompted the need to apply data-driven approaches to handle and make sense of learner data at scale [26]. Student essays, online forums, and online assignments are major venues and resources for large-scale text mining analysis to derive informative insights for evaluating performance or providing analytics insights for instructors and students to support learning[25]. NLP and machine learning algorithms are promising tools to automate this process. Previous studies have assessed discourse using text summarization [27], examined sociocognitive processes in collaborative conversations [21, 46], and modeled learner trajectories using neural network-based predictive methods [13].

Amongst an array of NLP techniques, topic modeling is an unsupervised method to extract emergent thematic structure in large textual data by connecting documents that share similar patterns[3, 34]. Topic modeling is deemed a powerful tool for education data mining and learning analytics research[48]. In empirical studies, topic modeling has been applied to examine course reviews to extract learner interests in order to provide personalized course recommendations to improve the quality of online courses[38, 42], to identify themes in asynchronous online discussions for providing adaptive support to individual students[23], or characterize chat dialogues within learning groups to support collaborative learning[12], or evaluating students' reflective writing to examine their learning experiences and engagement associated with course content[14].

Previously, research has pointed out discussion forums as an under-explored territory, where rich textual dialogue could offer the potential to support student learning[23]. While discussion forums may appear to be plagued by information overload and chaos, applying appropriate analytic techniques can help bring structure and identify important threads for students and teachers[56]. A growing body of learning analytics research focuses on discussion forum data to measure learner participation, and examine the associations between discourse behavior and learning[54]. Some studies leverage linguistic features such as sentiment valence to predict learning gains and retention for individual learners [55][37]. Other studies took a more exploratory perspective to investigate the temporal changes in MOOC learners' language and discourse characteristics at a high level [18]. As we have seen online discussion forums delivering promising results for social learning and student-teacher interactions in MOOCs[10], a question arises when a large number of courses at accredited universities started to rely on discussion forums during the pandemic: How can we leverage the large-scale, extensive data that emerged in the months before and after the outbreak of COVID-19 to understand discussion forum activities better? In fully remote instructions under the pandemic influence, the discussion forum is considered one of the most commonly used tools for supporting social interactions within online courses to meet the needs of a diverse student population[41]. Therefore, getting a better picture of the emergent topic and trends in this instructional environment would be beneficial to understanding how teaching and learning were affected during the critical months of the pandemic impact.

In investigating the impact of COVID-19, topic modeling has shown effectiveness in reflecting trends and public opinions on social media and online responses towards policy decisions[9, 47]. [39] studied the online mental health support community on Reddit (e.g., r/HealthAnxiety, r/schizophrenia) during the COVID-19 pandemic, looking into how different patterns of behavior can be captured through language and topic modeling. The increased use of discussion forum during the pandemic presents an opportunity to investigate learning activity in formal educational spaces. In particular, the timeline of COVID-19 spread and the academic calendar creates an unique alignment for potential changes in discussion forum discourse due to the influence of the pandemic and the transition to fully online learning. However, to our knowledge, no study has considered constructing temporal chains of topics in learner discourse that tracks these changes. Therefore, we seek to examine the rich textual data that exists in the Learning Management System (LMS) before and after universities shifted to remote learning.

2.2 Temporal Topic Modeling

With the ever-growing rate of data generation, topic modeling is a widely used approach to find patterns and trends within large-scale non-annotated data. Among several topic modeling techniques in NLP [34], one of the most widely used models is Latent Dirichlet Allocation (LDA) [8]. LDA is a generative statistical model that is used to extract latent topics from a text corpus. LDA models document as a probability distribution over topics where each topic is a probability distribution over words in the vocabulary. Recent advances in deep learning have introduced the combination of neural networks and transformer-based techniques [58, 28] to yield topics that are more coherent and interpretable than traditional models that use only bag-of-words (BoW) features. For instance, Combined Topic Model (CombinedTM) [5] is a recently proposed neural topic model that combines the bag of words (BoW) approach and the neural topic model ProdLDA [49] with the contextualized document representations from Sentence-BERT [45] for more coherent topics. CombinedTM is a Contextualized Topic Model (CTM) which uses Bag of Words (BoW) document representation concatenated with the contextualized document representations from Sentence-BERT [45] converted to the same dimensionality as of the BoW vocabulary by using a hidden neural network layer. This latent representation of the document is passed through a decoder network that reconstructs the BoW. The framework is originally based on a variational inference process [43]. Compared to existing topic models including the traditional LDA, CombinedTM showed more coherent topics and added contextual information [5]. In this study, we leverage CombinedTM to explore emerging topics and their evolution in learners' discussion forums before and during the onset of the COVID-19 pandemic.

Exploring how topics evolve can help us discover how certain events, such as the global health crisis, remote learning, and social injustice, impact learners' lived and academic experiences and shape their learning activities during the pandemic. Previous studies have proposed Dynamic Topic Model (DTM) [7] and related models [53] to detect and track topics over time-sequenced texts. However, these frameworks only model the variation in the probability distribution of words within a constant set of topics across time. To track how independent topics emerge, decline, and shift focus in a sequentially sliced corpus, researchers have used topic models and similarity metrics to connect topics in adjacent time steps [35, 31]. Similarity scores, such as Jaccard Coefficient, used in these studies do not account for the semantic similarity between words appearing in related contexts (e.g., college and university). To address this limitation, we propose a framework to detect and track semantically similar topics using WMD [36]. WMD measures the dissimilarity between two text documents, leveraging the power of word embeddings trained on a text corpus using the Word2Vec model [40]. We use this approach to find topics that represent a similar broad theme but depict a change in context in subsequent time steps. Notably, the WMD between two documents can be computed in a meaningful way even if the two documents do not have any words in common. WMD does not consider the order of words, making it suitable for tracking the similarity between two sets of words representing topics.

3. CURRENT STUDY

The objective of this study is to explore the emergent topics and their evolution in undergraduates' online discussion forums in the months prior to and after contingency shifts to fully remote learning due to the COVID-19 pandemic. Specifically, we aim to address the following questions:

> RQ1.a. What are the topics discussed by students before and after the pandemic?
> RQ1.b. How do these topics connect and evolve across months?
>
> RQ2: Do these topics represent discussions that are specific to certain courses or reflect general discourse across multiple courses?

To address the first research question, we used NLP techniques, specifically topic modeling, to capture the temporal

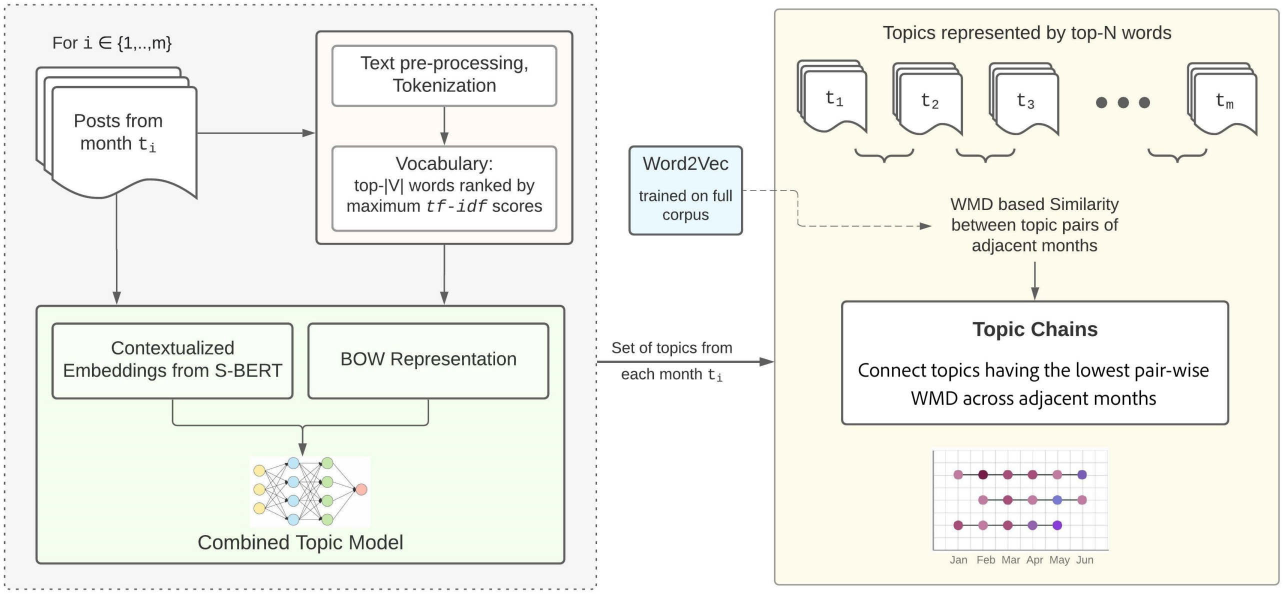

For i ∈ {1,..,m}
Posts from month t_i
Text pre-processing, Tokenization
Vocabulary: top-|V| words ranked by maximum tf-idf scores
Contextualized Embeddings from S-BERT
BOW Representation
Combined Topic Model
Set of topics from each month t_i
Word2Vec trained on full corpus
Topics represented by top-N words
t_1
t_2
t_3
t_m
WMD based Similarity between topic pairs of adjacent months
Topic Chains
Connect topics having the lowest pair-wise WMD across adjacent months
Jan
Feb
Mar
Apr
May
Jun

Figure 1: Graphical representation of the framework for topic modeling and topic evolution.

characteristics of learner discourse during this critical time in order to enhance our understanding of the influence of the pandemic on learning activities and learner discourse in a formal education setting. To achieve this goal, we introduce an analytical framework that comprises a CombinedTM [5] and a contextualized topic modeling (CTM) technique for extracting coherent themes that emerged in the discussion forum of the LMS across nine months. We then use Word Mover's Distance (WMD) to construct the linkage and temporal nuances of topics across months from October 2019 to June 2020 in order to build semantically meaningful topic chains that illustrate topical evolution over time. To address the second research question, we added a course-centricity measure to further contextualize whether a topic is specific to a course or discipline domain or more generic in the discussion forum.

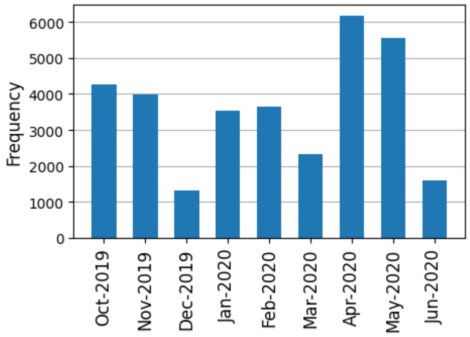


Figure 2: Frequency of posts in each month from October 2019 to June 2020.

Our current study contributes to the field of learning analytics by offering an analytic framework that is more interpretable for modeling topics in large-scale discourse data. The overview of our proposed framework is displayed in Fig. 1. We aim to extend the current literature by moving from mining static topics from large-scale student discourse to a more process-oriented perspective that shows how topics emerge, recur, and evolve over time. This analysis would allow us to view learner discourse as dynamic rather than a static picture and uncover relational linkages in the discourse.

4. DATA AND PARTICIPANTS

4.1 Participants

Our data were obtained from a large public university in the United States, which operates on a quarter system. We retrieved all discussion forum posts in the LMS across all courses offered in Fall 2019, Winter 2020, and Spring 2020 quarters (a total time span of nine months). We filtered the dataset to retain posts from a common set of students across each month (i.e., students who consistently contributed to the forum discussion). To eliminate individual differences, we focused on trends and themes of discourse originating from the same individuals. In addition, we considered posts that contained less than two words or had a length of fewer than five characters uninformative and removed them from the dataset. A total of 32,409 posts created by 449 students across 636 courses were retrieved, along with the time stamp of each post and the associated course and academic term. Given the rapid evolution of the pandemic, we use a month as the unit of analysis instead of the academic quarter to obtain more granular information about the temporal changes in student discourse. Fig. 2 shows the frequency of posts by month. We observe that term structure impacted the frequency of messages posted in discussion forums. Conclusion of courses before academic breaks, for example, winter break (Dec 2019) and summer break (Jun 2020), led to a lesser engagement in online discussions.

Discussion posts include conversations from 310 female students and 139 male students who identified with the following ethnic backgrounds: Asian/Asian American (49.7%), Hispanic (29.2%), White Non-Hispanic (13.6%), Black (4.0%), American Indian/Alaskan Native (0.2%), and others (3.3%). Overall, 52.5% of students identified as first-generation college students.

4.2 Pre-processing

To prepare the text for analysis, we pre-processed the data using Natural Language Toolkit (NLTK) [6] and Gensim [44], two open-source libraries in Python3. Website links and email addresses were replaced with "url" and "email" tokens, respectively. Contractions were fixed, and irrelevant and redundant terms such as punctuation, digits, and stop-words were removed. To retain relevant words for building topic models, we used the Part-of-Speech Tagger to identify nouns, verbs, adjectives, and adverbs, which were further lemmatized using the WordNetLemmatizer from the NLTK package. Lemmatization retains the base word known as lemma (e.g., study) from inflected forms of a word (e.g., studying, studies, studied).

5. TOPIC DETECTION AND TRACKING

5.1 Topic Detection

Regarding RQ1.a, we first detected the emerging topics in the student's discussion forum data. To achieve this, we trained CombinedTM on the messages posted by students over each month separately. The quality and interpretability of the resulting topics depend on the curated vocabulary [24]. We constructed the BoW vocabulary by retrieving the top 10,000 words with maximum TF-IDF (Term Frequency - Inverse Document Frequency) weights. Sentence-BERT [45], a modification of the BERT [16] model, was used to obtain meaningful encodings of the messages. To optimize the number of topics (K) as a hyperparameter for each month, we ran the model on the corpus of each month with K ranging from 5 to 15 topics heuristically. While having less than five topics would result in overly generic and less coherent topics, on the other hand, a large number of topics causes highly coherent topics with lower topic diversity. Therefore, to optimize the number of topics, it is necessary to evaluate them on the three metrics used in CombinedTM that evaluate topic coherence and topic diversity:

1. Normalized pointwise mutual information
2. Word embeddings-based similarity
3. Inversed Rank-Biased Overlap

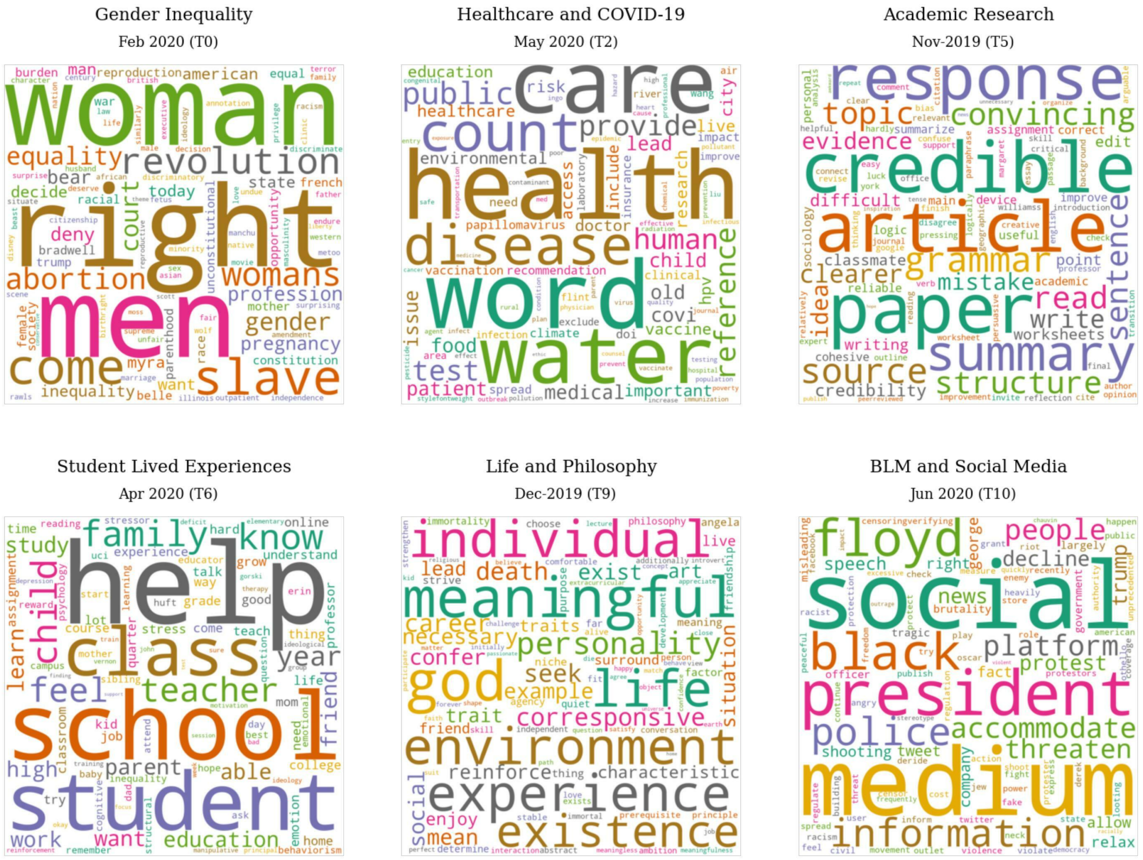
Normalized pointwise mutual information (NPMI) was used to measure how related the top-10 words of a topic are to each other, considering the words' empirical frequency in the original corpus. Word embeddings-based similarity refers to the average pairwise cosine similarity of the word embeddings of the top-10 words in a topic, using Word2Vec [40] word embeddings. The overall average of those values for all the topics was computed. Inversed Rank-Biased Overlap (IRBO) was used to score the diversity of the resulting topics, and NPMI and Word embeddings-based similarity were used to measure the average topic coherence. Subsequently, we used the soft voting ensemble approach to select the optimal K corresponding to the model which returned the highest sum of the normalized values of these metrics.

5.2 Topic Evolution

Previous research on topic tracking uses similarity metrics such as Kullback-Leibler (KL) divergence, Jaccard Coefficient, Kendall's Coefficient, and Cosine similarity between the probability distribution vectors of two topics. However, these metrics do not account for the semantic similarity between words of the same theme (e.g., student and university). Hence, in order to measure the semantic and contextual similarity between topics, we use the Word Mover's Distance (WMD). If there is a connection between two topics in consequent months, we consider that as an evolution of a topic and call it a "Topic Chain". More specifically, to address RQ1.b, we used the WMD to measure the semantic similarity between two topics. WMD finds an optimal solution to a transportation problem, which determines the minimum cost to move all words from one document to another. We trained a Word2Vec model using the skip-gram algorithm over the discussion posts data with a sliding window size of five, to obtain a lower-dimensional representation ($d = 100$) of words present in the entire corpus. Next, we represented each topic as a list of top-30 most frequent words in that topic and computed the WMD between two topics ϕ_i^t and ϕ_j^{t+1} for every i, j where $i \in \{0, ..., K_t\}$, $j \in \{0, ..., K_{t+1}\}$ and $t, t+1$ represent two consecutive months. For every topic ϕ_i^t in month t, we selected the topic from month $t+1$ which had the least WMD from it, thereby denoting the highest similarity. To keep a record of the connected topics, we created a mapping from ϕ_i^t to ϕ_j^{t+1} and saved the corresponding WMD. To avoid multiple topics in month t being mapped to the same topic in month $t+1$, we retained only the topic pairs having the least WMD among them. We created a directed graph connecting nodes (or topics) in consecutive months. Finally, we found all the simple paths from each root to leaf in this graph using the NetworkX package [29]. These directed paths are referred to as "topic chains". Effectively, we identified the most semantically similar topic pairs in subsequent months and connected them to construct topic chains.

5.3 Course-centricity of Topics

As noted above, discussion posts were taken from over 600 different courses. Inevitably, the content of each post can be heavily influenced by the course. Some discussions may be prompted by the course instructor or students, which as a result could impact the discussion forum conversations and the organic flow of the discussions. Consequently, these course-specific combined with several related courses could result in detected topics centered around course-related material. By contrast, other topics might represent a theme of discussions that originate from and are present in a broader range of courses, which makes the topic less dependent on course material resulting in more "general" topics. To address RQ2, we propose a measure of course-centricity to distinguish topics that are more specific to certain courses from topics that represent a broader theme that emerged repeatedly across mul-



Gender Inequality

Feb 2020 (T0)

Healthcare and COVID-19

May 2020 (T2)

Academic Research

Nov-2019 (T5)

Student Lived Experiences

Apr 2020 (T6)

Life and Philosophy

Dec-2019 (T9)

BLM and Social Media

Jun 2020 (T10)

Figure 3: Word-clouds of selected topics. The size of words is directly proportional to the topic-term probability, where a larger size represents higher relevancy to the topic.

tiple courses. We associated the course-centricity of a topic with the standard deviation of the distribution of message counts across courses. For each discussion post, we selected the topic with the highest probability of being assigned as the main topic of the post. Next, we selected the top ten courses with the most number of posts in each topic. The frequencies of the posts corresponding to the top-$N(=10)$ most common courses were used to calculate the standard deviation (σ) of the distribution of posts for each topic. A lower value of σ denoted a relatively uniform distribution of courses in a topic, suggesting a topic is more generally distributed across courses. A higher value of σ denoted a skewed distribution where very few courses dominate the discussion, showing that the topic is more "course-centric". In addition, we measured the number of messages in each of the top ten courses for the topic. Topics that were mostly course-centric would have a disproportionately higher message count from one or two courses than from other courses, while more general topics would have a more even distribution of message count across the top ten courses.

6. RESULTS AND DISCUSSION

6.1 Topic Detection

The topic modeling resulted in 8-13 topics per month, including literature, philosophy, global health, and COVID-19-related discourse. We also observed that casual discourse and discussions on academic work remained the most common topics across all months. This implies that discussion forum participation includes not only learners' active knowledge construction that correlates with grades and learning outcomes, but also social connectedness that might affect learning experience online. Topics such as immigration and ethnic diversity, student lived experiences, social effects of technology and socializing in school revealed a deeper insight into students' personal views. Moreover, social justice movements such as the Black Lives Matter movement during the spring of 2020 emerged as a novel topic in discussion and conversation among students. This signals the influence of contemporary events on learning, and that students are actively trying to make sense of what is happening in the world and integrating that reflection into learning. Fig. 3 demonstrates the word clouds of selected topics discussed below. The topics are inferred by the authors from the distribution of words and posts representing them. A list of top-ranked words representing all the identified topics and their corresponding labels interpreted by the authors are publicly available[1].

[1] github.com/The-Language-and-Learning-Analytics-Lab

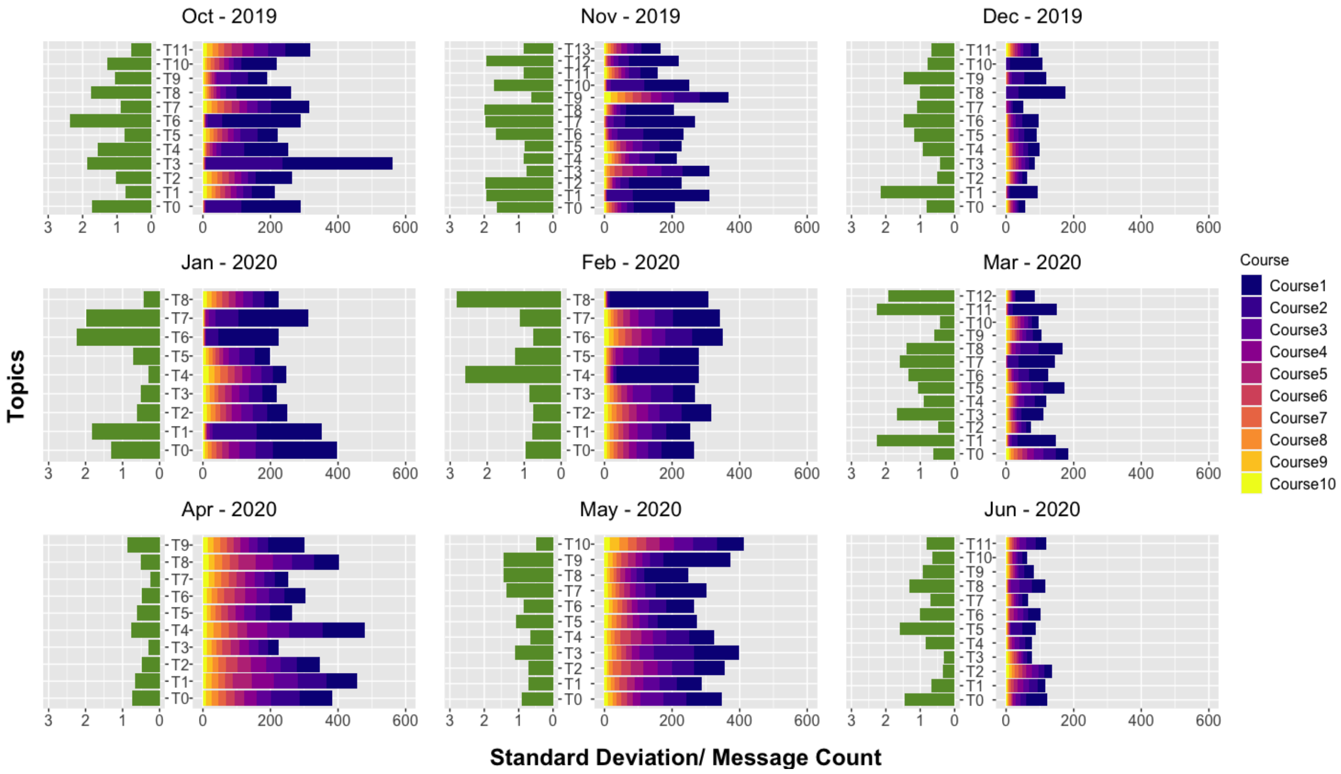


Figure 4: Degree of course-centricity of topics across nine months. Colored gradient bars represent the number of messages in each of the top ten courses that contributed to the topic. Green bars represent the standard deviation of messages from the top ten courses.

6.2 Course-centricity of Topics

In Fig. 4, we demonstrate the course-centricity of topics. Course-centric topics are represented by fewer variations in colors in the gradient bars and a greater standard deviation. By contrast, a more uniform distribution of colors and lower standard deviations such as Socio-cultural Deviance (T2) and Miscellaneous Discourse (T7) in April 2020 demonstrate that a topic is more generally distributed across various courses. We note two key findings. First, we found causal conversations were more generally distributed across courses while topics such as Global Health (T6) and Social Inequities (T7) in January 2020 were more course-centric. Our results suggest a combination of standard deviation and message count can successfully capture variations in topics' course-centricity. Second, we observed a decrease in course-centric topics in the Spring quarter compared to Fall and Winter quarters, which suggests a shift in students' conversations as they transitioned to online learning. To empirically test this shift, we performed a post-hoc Welch's Two Sample t-test to compare the standard deviation of Fall and Winter quarters with that of the Spring quarter. Standard deviations for the two groups differed significantly ($t(5.36)$ $= p < .001$) such that combined, Fall and Winter quarters had a greater standard deviation (M=.10) than in Spring quarters (M=.03). Notably, the Spring quarter began a few weeks after COVID-19 was declared a pandemic and fully remote learning was implemented. This finding suggests a general change in the online discussion forum landscape, from supporting course-centric content discussion prior to March 2020, to an increased presence of social interactions and discussion on shared live experiences. This might indicate students seeking opportunities to engage in casual interaction that used to take place in the hallways, walking to classes, after classes, or instructors' attempt in facilitating social presence and creating classroom community as well as integrating critical reflections between learning material and contemporary events during remote learning.

6.3 Topic Evolution

Topic chains constructed using WMD illustrate the evolving and emerging topics (Fig. 5). Our findings reveal two topic chains (Chains 12-13) that are consistently present throughout all nine months of the academic year. Some topic chains are limited to a specific quarter (Chains 1-4) which may reflect the classes offered during that quarter. In contrast, other chains notably stop at (Chains 8-10) or start during (Chains 4 and 6) the transition to online learning which may signal the influence of the pandemic on the topics students discussed. We discuss each of these in turn.

Consistently present, Chain 12 began with Public Health-related discussions in October 2019. This topic chain subsequently linked topics of Health Issues, Healthcare & Social Inequities, and Healthcare & Government. In March 2020, this topic evolved into discussions on the Public Health of China. Notably, this shift occurred at the time COVID-19 was declared a pandemic. These themes further transitioned to Public Health Inequities and eventually COVID-19 related discourse in the Spring quarter. Chain 12 also demonstrates that the topics in earlier months were more

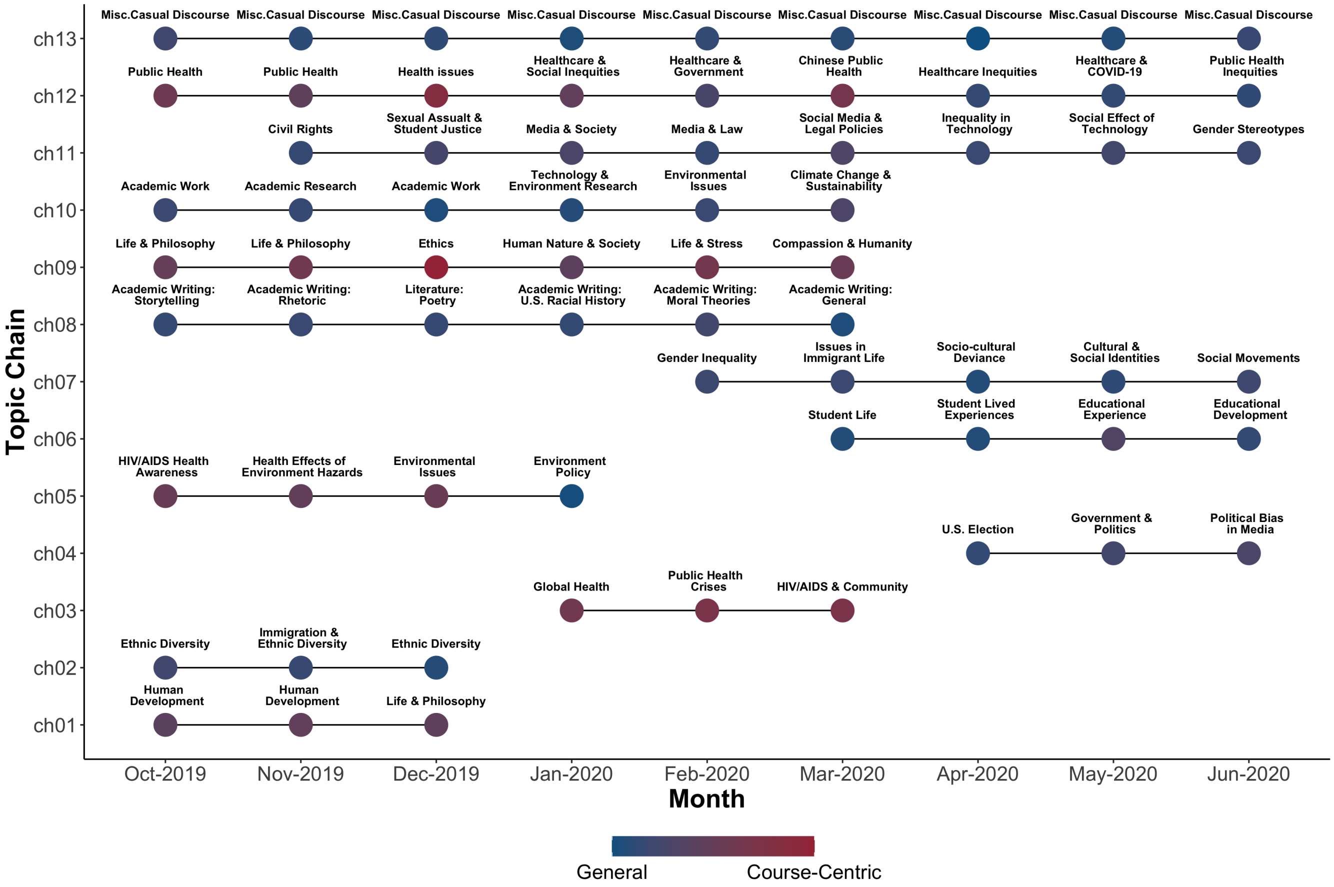


Figure 5: Topic chains identified using the proposed framework. The heat-map scale denotes the course-centricity of a topic.

course-centric, but starting in April 2020 there was a shift towards more general discourse regarding public health inequities and the COVID-19 pandemic in various courses. Another long topic chain (Chain 13) represents miscellaneous messages and casual interactions across all months. This consistent presence implies that online discussions contained some level of student casual interactions prior to and at the onset of the pandemic.

Some topic chains either stopped or began during the onset of the pandemic. For example, Academic writing (Chain 8) and Academic Work (Chain 10) and their respective subtopics were connected from October 2019 to March 2020. These chains highlight the various forms of essay and prompt responses, and reactions to readings or weekly discussion posts that are often found in online discussion classes. Notably, both of these chains contained more "general" topics. We note two possible explanations for the chains' drop-off in March 2020. First, this drop-off may suggest that the courses that prompted these discussions were either no longer available to students or fewer students signed up for these courses during the Spring quarter. Second, it is possible that these discussions surrounding academic work and writing were still taking place, but outside of online discussion forums (e.g., virtual groups, breakout rooms online). Similarly, Chain 9, "Life and Philosophy" related discourse stopped around March 2020. Although most of the posts under these topics were induced by course-related prompts, a closer inspection revealed many posts where students communicated personal experiences and thoughts with their peers. The drop-off for more course-centric chains like Chain 9 could also be due to courses no longer available to students, or perhaps to fewer discussion-based assignments in these courses.

Student life and student's lived experiences were identified as relatively new topics starting in March 2020 (Chain 6). Students' posts included a variety of university-related experiences and major family and life events. A rise in such posts among students during the switch to a fully online education system demonstrated an evolved use of discussion forums to connect with peers and express their thoughts and concerns. These topics were further connected to topics representing shifts in educational experiences. Students expressed their struggles in coping with the sudden transition to online classes and the rapid spread of COVID-19 around the world which added personal or financial difficulties. Other notable posts under this topic included activities that helped them deal with stressful times, their study regime, and new performance evaluation strategies. Other prominent topic chains include Chain 11, "Civil rights" which evolved into related discussions around sexual assault, law and legal policies associated with social media, and social and gender inequalities. Gender inequality also emerged as a topic in Chain 7 and was linked to other social cause issues regarding immigration, cultural and social movements. Social movements is a diverse topic including terms and events related to feminism, political influence, bias in disseminated information, and fake news.

6.4 Limitations

Our findings demonstrate a clear influence of the COVID-19 pandemic and online learning activities. However, there are a few limitations to our study. The unsupervised nature of topic modeling requires some subjectivity where the authors create topic labels by interpreting the top-ranked words. Future studies should consider blinding the researcher from the month's name when viewing top-ranked words to label topics. Although it is a common practice, the topic labeling could be consequently influenced by the authors' preconceptions of the impact of the pandemic on students' discourse. Moreover, our study does not capture whether a topic is initiated by students or instructors. Although a topic with high course-centricity is influenced by the instructor's prompts to discuss specific academic topics, in some cases, professors also direct and facilitate casual interactions (i.e., introducing themselves, or sharing their experiences during the pandemic). Lastly, our findings reflect discourse changes within asynchronous interactions in online discussions and cannot infer any discourse that took place in synchronous classroom settings or in classes that did not utilize discussion forums for instruction.

6.5 Implications and Future Directions

Large-scale educational data from online discussion forums is an under-excavated gold mine for understanding learning behavior. Educational theories such as constructivism emphasize the role of social interactions and the construction of knowledge through experience and reflection. Analyzing educational discourse can help researchers understand the process of learners' sense-making of new concepts, as well as their attitudes, engagement levels, and experiences.

Extracting meaningful insights from unstructured educational interactions and discussions requires accurate content representation and context consideration. Modeling discourse structure and dynamics pose challenges, such as identifying key topics, tracking the evolution of ideas, and capturing the social and emotional dimensions of communication. To address these challenges, our paper suggests a solution that includes additional LMS features for meaningful interpretation, and effective NLP methods for capturing the dynamic of discourse across different domains and contexts. Specifically, we added LMS features (i.e. message count and standard deviation) to characterize the course-centricity feature of topics to strengthen interpretability. With the case of tracking themes before and after the COVID-19 pandemic, we prove that CTM and WMD can be effective tools to capture emergent topics over time.

This framework is a flexible and adaptable tool that can be adapted to explore other educational research questions in different contexts to investigate teaching and learning behavior in online environments. For instance, future studies could use this method to examine how discourse evolves within specific disciplines (e.g., business courses, STEM courses), potentially for monitoring the consistency of course discourse space. If an instructor wants to understand how students' discourse has changed for a repeatedly offered foundational course, this analytical framework will also be effective for revealing the discourse evolution from past to present to provide insights for the instructor on curriculum design. Future research might also apply it to other formal and informal learning contexts, i.e., MOOCs, social media discussion, and by adding LMS features relevant to self-regulation, which may provide meaningful interpretations of the discourse pattern. While our study only focused on tracking the topics for students who had consistently contributed to the discussion forum, future research may consider how topics evolve across subgroups of students with different demographic backgrounds and individual characteristics such as learner motivation. Although exploring the origins of topics is beyond the scope of this paper, it would be interesting for future research to investigate whether there are differences between conversations initiated by instructors versus students.

7. CONCLUSION

We used an NLP-driven topic detection and tracking approach to detect emergent topics and model the evolution of various topics in students' online academic discourse over time. We demonstrate how this novel NLP technique can be used to provide meaningful insights on large-scale unstructured student data from discussion forums effectively. Our study contributes to the current literature by moving beyond mining static topics from large-scale student discourse to a more process-oriented, temporal lens on how topics emerge, recur and evolve over time. We used contextualized topic models to identify coherent topics from each month, which are more interpretable than traditional models that use only bag-of-words (BoW) features. Some of the identified topics were found to be originating from discussions pertaining to specific courses, while other topics demonstrated the expression of personal opinions and beliefs. Using standard deviation to identify course-centricity, we found that topics posted at the beginning of the pandemic were relatively more general than those before March 2020. This significant increase in casual interactions since Spring 2020 indicates a shift in the discussion forum's function from predominantly enabling academic discourse to increasingly facilitating peer interactions. This evidence supports claims from previous studies that instructors across disciplines were leveraging discussion forums to support social interactions and social learning. The emergent discussion surrounding COVID-19 and other contemporary events in learner discourse suggests that their impacts on students learning and lived experience are not mutually exclusive and are exhibited in both academic and casual discourse. Furthermore, by denoting topics as a set of top representative words based on topic-term probability, we computed the Word Mover's Distance as a semantic similarity metric between adjacent months' topics. The most similar topics were connected and topic chains were constructed to uncover their evolution and identify newer themes. For researchers and practitioners in the EDM community, our proposed approach provides a viable means to characterize topic trends over time in learner discourse at different granularity, such as for specific courses or other online learning contexts. This method might also be generalized to other types of educational discourse to detect and track specific policy impacts or instructional interventions on students' online discussion activities. Lastly, although this study focused specifically on the events during the 2019-2020 academic year, this approach could be further utilized to understand the temporal dynamics of discussion data in broader contexts and timeframes, i.e., MOOC discussions and social media data.

8. REFERENCES

[1] H. E. Abdelkader, A. G. Gad, A. A. Abohany, and S. E. Sorour. An efficient data mining technique for assessing satisfaction level with online learning for higher education students during the covid-19. *IEEE Access*, 10:6286–6303, 2022.

[2] A. Ahadi, A. Singh, M. Bower, and M. Garrett. Text mining in education—a bibliometrics-based systematic review. *Education Sciences*, 12(3):210, 2022.

[3] R. Alghamdi and K. Alfalqi. A survey of topic modeling in text mining. *Int. J. Adv. Comput. Sci. Appl.(IJACSA)*, 6(1), 2015.

[4] O. Almatrafi and A. Johri. Systematic review of discussion forums in massive open online courses (moocs). *IEEE Transactions on Learning Technologies*, 12(3):413–428, 2018.

[5] F. Bianchi, S. Terragni, and D. Hovy. Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2021.

[6] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[7] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 113–120. Association for Computing Machinery, 2006.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[9] S. Boon-Itt, Y. Skunkan, et al. Public perception of the covid-19 pandemic on twitter: sentiment analysis and topic modeling study. *JMIR Public Health and Surveillance*, 6(4):e21978, 2020.

[10] C. G. Brinton, M. Chiang, S. Jain, H. Lam, Z. Liu, and F. M. F. Wong. Learning about social learning in moocs: From statistical analysis to generative model. *IEEE transactions on Learning Technologies*, 7(4):346–359, 2014.

[11] D. Bylieva, Z. Bekirogullari, V. Lobatyuk, and T. Nam. Analysis of the consequences of the transition to online learning on the example of mooc philosophy during the covid-19 pandemic. *Humanities & Social Sciences Reviews*, 8(4):1083–1093, 2020.

[12] Z. Cai, B. Eagan, N. Dowell, J. Pennebaker, D. Shaffer, and A. Graesser. Epistemic network analysis and topic modeling for chat data from collaborative learning environment. In *Proceedings of the 10th international conference on educational data mining*, 2017.

[13] C. Chen and Z. Pardos. Applying recent innovations from nlp to mooc student course trajectory modeling. *arXiv preprint arXiv:2001.08333*, 2020.

[14] Y. Chen, B. Yu, X. Zhang, and Y. Yu. Topic modeling for evaluating students' reflective writing: a case study of pre-service teachers' journals. In *Proceedings of the 6th International Conference on Learning Analytics and Knowledge*, 2016.

[15] H. Chopra, Y. Lin, M. A. Samadi, J. G. Cavazos, R. Yu, S. Jaquay, and N. Nixon. Modeling student discourse in online discussion forums using semantic similarity based topic chains. In M. M. Rodrigo, N. Matsuda, A. I. Cristea, and V. Dimitrova, editors, *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners' and Doctoral Consortium*, pages 453–457, Cham, 2022. Springer International Publishing.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

[17] N. Dowell and V. Kovanovic. Modeling educational discourse with natural language processing. *education*, 64:82, 2022.

[18] N. M. Dowell, C. Brooks, V. Kovanović, S. Joksimović, and D. Gašević. The changing patterns of mooc discourse. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, pages 283–286, 2017.

[19] N. M. Dowell, A. C. Graesser, and Z. Cai. Language and discourse analysis with coh-metrix: Applications from educational material to learning environments at scale. *Journal of Learning Analytics*, 3(3):72–95, 2016.

[20] N. M. Dowell, Y. Lin, A. Godfrey, and C. Brooks. Exploring the relationship between emergent sociocognitive roles, collaborative problem-solving skills, and outcomes: A group communication analysis. *Journal of Learning Analytics*, 7(1):38–57, 2020.

[21] N. M. Dowell, T. M. Nixon, and A. C. Graesser. Group communication analysis: A computational linguistics approach for detecting sociocognitive roles in multiparty interactions. *Behavior research methods*, 51(3):1007–1041, 2019.

[22] N. M. M. Dowell and A. C. Graesser. Modeling learners' cognitive, affective, and social processes through language and discourse. *Journal of Learning Analytics*, 1(3):183–186, 2014.

[23] A. Ezen-Can, K. E. Boyer, S. Kellogg, and S. Booth. Unsupervised modeling for understanding mooc discussion forums: a learning analytics approach. In *Proceedings of the fifth international conference on learning analytics and knowledge*, pages 146–150, 2015.

[24] A. Fan, F. Doshi-Velez, and L. Miratrix. Assessing topic model relevance: Evaluation and informative priors. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 12(3):210–222, 2019.

[25] R. Ferreira-Mello, M. André, A. Pinheiro, E. Costa, and C. Romero. Text mining in education. *Wiley Interdisciplinary Reviews: Data Mining and*

Knowledge Discovery, 9(6):e1332, 2019.

[26] C. Fischer, Z. A. Pardos, R. S. Baker, J. J. Williams, P. Smyth, R. Yu, S. Slater, R. Baker, and M. Warschauer. Mining big data in education: Affordances and challenges. *Review of Research in Education*, 44(1):130–160, 2020.

[27] F. B. Goularte, S. M. Nassar, R. Fileto, and H. Saggion. A text summarization method based on fuzzy rules and applicable to automated assessment. *Expert Systems with Applications*, 115:264–275, 2019.

[28] M. Grootendorst. Bertopic: Leveraging bert and c-tf-idf to create easily interpretable topics., 2020.

[29] A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using networkx. In *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, 2008.

[30] L. Hakimi, R. Eynon, and V. A. Murphy. The ethics of using digital trace data in education: A thematic review of the research landscape. *Review of educational research*, 91(5):671–717, 2021.

[31] F. Jian, W. Yajiao, and D. Yuanyuan. Microblog topic evolution computing based on lda algorithm. *Open Physics*, 16(1):509–516, 2018.

[32] S. Joksimović, N. Dowell, O. Poquet, V. Kovanović, D. Gašević, S. Dawson, and A. C. Graesser. Exploring development of social capital in a cmooc through language and discourse. *The Internet and Higher Education*, 36:54–64, 2018.

[33] Z. Kanetaki, C. I. Stergiou, G. Bekas, C. Troussas, and C. Sgouropoulou. Data mining for improving online higher education amidst covid-19 pandemic: A case study in the assessment of engineering students. In *NiDS*, pages 157–165, 2021.

[34] P. Kherwa and P. Bansal. Topic modeling: A comprehensive review. *EAI Endorsed Transactions on Scalable Information Systems*, 7(24), 2019.

[35] D. Kim and A. H. Oh. Topic chains for understanding a news corpus. In *Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing - Volume Part II*, 2011.

[36] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, 2015.

[37] Y. Lin, R. Yu, and N. Dowell. Liwcs the same, not the same: Gendered linguistic signals of performance and experience in online stem courses. In *International Conference on Artificial Intelligence in Education*, pages 333–345. Springer, 2020.

[38] S. Liu, C. Ni, Z. Liu, X. Peng, and H. N. Cheng. Mining individual learning topics in course reviews based on author topic model. *International Journal of Distance Education Technologies (IJDET)*, 15(3):1–14, 2017.

[39] D. M. Low, L. Rumker, T. Talkar, J. Torous, G. Cecchi, and S. S. Ghosh. Natural language processing reveals vulnerable mental health support groups and heightened health anxiety on reddit during covid-19: Observational study. *Journal of medical Internet research*, 22(10):e22635, 2020.

[40] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[41] A. Naim. Application of digital technologies for the students with diverse skills during covid: 19. *American Journal of Research in Humanities and Social Sciences*, 1:46–53, 2022.

[42] G. Nanda, K. A. Douglas, D. R. Waller, H. E. Merzdorf, and D. Goldwasser. Analyzing large collections of open-ended feedback from mooc learners using lda topic modeling and qualitative analysis. *IEEE Transactions on Learning Technologies*, 14(2):146–160, 2021.

[43] R. Ranganath, S. Gerrish, and D. Blei. Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR, 2014.

[44] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50. ELRA, 2010.

[45] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019.

[46] M. A. Samadi, J. G. Cavazos, Y. Lin, and N. Nixon. Exploring cultural diversity and collaborative team communication through a dynamical systems lens. 2022.

[47] H. Sha, M. A. Hasan, G. Mohler, and P. J. Brantingham. Dynamic topic modeling of the covid-19 twitter narrative among us governors and cabinet executives. *arXiv preprint arXiv:2004.11692*, 2020.

[48] S. Slater, S. Joksimović, V. Kovanovic, R. S. Baker, and D. Gasevic. Tools for educational data mining: A review. *Journal of Educational and Behavioral Statistics*, 42(1):85–106, 2017.

[49] A. Srivastava and C. Sutton. Autoencoding variational inference for topic models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[50] W. Strielkowski. Covid-19 pandemic and the digital revolution in academia and higher education. 2020.

[51] C.-H. Tu. On-line learning migration: From social learning theory to social presence theory in a cmc environment. *Journal of network and computer applications*, 23(1):27–37, 2000.

[52] L. S. Vygotsky and M. Cole. *Mind in society: Development of higher psychological processes.* Harvard university press, 1978.

[53] C. Wang, D. Blei, and D. Heckerman. Continuous time dynamic topic models. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, UAI'08, page 579–586. AUAI Press, 2008.

[54] X. Wang, D. Yang, M. Wen, K. Koedinger, and C. P. Rosé. Investigating how student's cognitive behavior in mooc discussion forums affect learning gains. *International Educational Data Mining Society*, 2015.

[55] M. Wen, D. Yang, and C. Rose. Sentiment analysis in mooc discussion forums: What does it tell us? In

Educational data mining 2014. Citeseer, 2014.

[56] A. F. Wise, Y. Cui, and J. Vytasek. Bringing order to chaos in mooc discussion forums with content-related thread identification. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 188–197, 2016.

[57] World Health Organization. WHO Director-General's opening remarks at the media briefing on COVID-19 - 11 March 2020., 2020. https://www.who.int.

[58] H. Zhao, D. Phung, V. Huynh, Y. Jin, L. Du, and W. Buntine. Topic modelling meets deep neural networks: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4713–4720, 2021.

Analysis of an Explainable Student Performance Prediction Model in an Introductory Programming Course

Muntasir Hoq
North Carolina State University
Raleigh, NC
mhoq@ncsu.edu

Peter Brusilovsky
University of Pittsburgh
Pittsburgh, PA
peterb@pitt.edu

Bita Akram
North Carolina State University
Raleigh, NC
bakram@ncsu.edu

ABSTRACT

Prediction of student performance in Introductory programming courses can assist struggling students and improve their persistence. On the other hand, it is important for the prediction to be transparent for the instructor and students to effectively utilize the results of this prediction. Explainable Machine Learning models can effectively help students and instructors gain insights into students' different programming behaviors and problem-solving strategies that can lead to good or poor performance. This study develops an explainable model that predicts students' performance based on programming assignment submission information. We extract different data-driven features from students' programming submissions and employ a stacked ensemble model to predict students' final exam grades. We use SHAP, a game-theory-based framework, to explain the model's predictions to help the stakeholders understand the impact of different programming behaviors on students' success. Moreover, we analyze the impact of important features and utilize a combination of descriptive statistics and mixture models to identify different profiles of students based on their problem-solving patterns to bolster explainability. The experimental results suggest that our model significantly outperforms other Machine Learning models, including KNN, SVM, XGBoost, Bagging, Boosting, and Linear regression. Our explainable and transparent model can help explain students' common problem-solving patterns in relationship with their level of expertise resulting in effective intervention and adaptive support to students.

Keywords

Explainable student modeling, Student programming analysis, Student programming pattern, Student performance prediction, Student profiling

1. INTRODUCTION

Introductory programming courses (CS1) have been observing a constant surge in interest and enrolment of students in recent years [38]. With this growing interest in Computer Science, the number of students struggling and dropping out of courses is also increasing [16, 37, 49]. Automated prediction systems can help to prevent this by predicting student performances and enabling instructors to intervene effectively [20, 46, 52].

Early methods to predict student performances in CS1 exploited different static approaches based on student starting data such as age, gender, grades, etc. [1]. However, predicting student performance statically is challenging as their behaviors are dynamic that can change with time [37, 45]. More recently, research on performance prediction focused on data-driven approaches incorporating Machine Learning (ML) techniques [17, 22, 40, 35]. However, most of these approaches do not analyze students' programming behaviors focusing instead on intermediate assessment data such as quiz scores and midterm exam grades. It prevents these methods from understanding the source of student problems and generalizing over different CS1 courses. Moreover, many courses may not even include any interim exams, as in the case of the dataset used in this study. Moreover, the explainability and transparency of black box ML models are becoming as important as high predictive power. An explainable model can help instructors and students understand the predictions and gain more trust. It can enable instructors to gain insights into students' problem-solving strategies by understanding the patterns of different students' programming behaviors. It can also help in effective intervention to help struggling students in the learning process. There are already a few studies that explore explainable performance prediction models in the field of Education [35, 9]. However, to the best of our knowledge, no other study has employed explainable models to analyze student performance based on students' programming behaviors without considering exam or quiz grades.

In this study, we propose an explainable student performance prediction model that can predict students' final exam grades from their programming assignment submission data. Predicting final exam grades only from programming assignments is a challenging task since the nature of the final exam can differ from the assignments. We employ a data-driven feature extraction approach to select features representing students' programming behaviors in a CS1 course. We develop a stacked ensemble regression model to predict students' final exam grades. Our stacked ensemble model has KNN, SVM, and XGBoost as the base models and Lin-

M. Hoq, P. Brusilovsky, and B. Akram. Analysis of an explainable student performance prediction model in an introductory programming course. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 79–90, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115693

ear regression as the meta-model. We compare the performance with other baseline techniques, including the individual components of our stacked ensemble model: Linear regression, KNN, SVM, and XGBoost, and other ensemble techniques such as Bagging and Boosting. The experimental results suggest that our model significantly outperforms these baseline techniques. Furthermore, we employ SHAP [27], a game-theory-based framework, to explain our model's predictions based on the importance and impacts of features. We explain students' performance predictions to understand how each feature contributes to the prediction process of students' final exam grades at an individual student level and a global level with all the students. This enables us to analyze students' performance predictions based on their underlying programming behaviors. We also analyze important features and utilize a combination of descriptive statistics and mixture modeling to understand student patterns of behavior. This provides insights for the instructors into different profiles of students' learning progressions to make informed decisions about intervening with struggling students and provide adaptive support [4].

The main contributions of this study are as follows:

- Building an explainable stacked ensemble model to predict the student performance in the final exam using programming assignment data of students.
- Explaining the predictions of the model at an individual and a global level of different programming information to gain the trust of the stakeholders.
- Analyzing the results of SHAP and important features of the explainable model to profile students based on their behavior and gain insight into their problem-solving strategies and connection to their learning outcomes.

2. RELATED WORK

In this section, we explore different techniques and studies done in the field of student performance prediction and the use of explainable models in programming.

2.1 Student Performance Prediction

Predicting struggling students and their success has been an important area for researchers in intelligent tutoring systems. These studies have different goals, i.e., predicting students' early success, detecting failing students, detecting dropouts at an early stage, predicting student performance in the final exam, etc.

A systematic review of previous research on student performance predictions was conducted in [40]. The review revealed that most of these studies used features such as cumulative grade point averages (CGPA) and other assessments (quizzes, midterms, etc.). In [22], an open-source predictive platform was developed and used in the at-risk student detection task. The data included demographic and enrollment information and was classified using ML models such as SVM and Linear Regression. In a recent study [17], enrolled students were classified into passing and failing categories using Decision Tree and SVM from features such as quizzes and midterm exam scores. Recognition of at-risk students could be used for early intervention.

Features like weekly assignment scores, midterm exam grades, etc., were used in [10] to predict failing students in an introductory programming course. Another study [21] followed a similar feature set and employed different ML algorithms to verify their effectiveness in student performance prediction. On event-level analysis, such as predicting students' success in completing programming exercises, [28] used Recent Temporal Patterns and LSTM. Another study [36] predicted early dropout of students for online programming courses. They used features from online platforms, such as student login times, keystroke latency, correctness, etc., for the first time. In [35], students' early performance was predicted for an introductory programming course from their midterm exam grade, procrastination time, correctness, the total number of logical lines in code, copy-paste information, etc., from an online programming system using XGBoost.

Recently, different Deep Learning frameworks have been noticeably used in students' success prediction and are increasing. In a recent study [14], an abstract syntax tree (AST)-based embedding model, SANN, showed effectiveness in capturing information from student programming codes. In [51, 30], abstract syntax tree-based and control flow graph-based embedding models were used to predict students' final exam grades from their programming assignment data. In another recent study [5], CNN and LSTM networks, along with programming code submission metadata, were used to predict student performance on the final exam in an introductory programming course.

However, previous studies proved effective in student performance prediction tasks; we identify different challenges associated with these studies. Static approaches fail to capture the dynamic behavior of students during their learning process. Data-driven approaches followed in prior studies are not generalizable in different programming courses since they differ in the course outline, and interim exams can vary from course to course. Moreover, some introductory programming courses might not include any interim exam at all, as the case with the dataset used in this study, where only programming assignments are available. Furthermore, deep-learning models are becoming popular with time; however, it is challenging to achieve good performance with these models trained on such small classroom-sized datasets [29].

2.2 Explainable Artificial Intelligence (XAI)

XAI helps humans understand a black-box ML model. It interprets a model's outcomes and explains the reasons behind decisions. XAI algorithms have been extensively used in different areas of research as well as in medical, health care, and clinical data analysis [33, 44, 19], industrial data analysis [2, 39], smart city solutions [47, 12], etc.

Though XAI is becoming a popular approach to interpreting and explaining ML solutions, its effectiveness is relatively unexplored in CS Education and intelligent tutoring systems. In [25], a deep learning-based knowledge tracing model was developed. The model was interpreted using the layer-wise relevance propagation method. A recent work [43] incorporated explainable concepts into computational models for student modeling tasks in computing education; while the work has been exploratory, the performance needs to be further improved for actual deployment. Mu et al. [32] automatically informed the individualized intervention by detecting wheel-spinning students, where students try and repeatedly fail at an educational task based on the number of attempts. Shapley values were used to explain the outcomes of the ML models used, including Linear Regression and XGBoost. In another study [6], university dropout prediction was made using a fully connected deep neural network. The features used in this study included university program-related data, high school performance-related data, matura exam (an exam after secondary school) results, average exam grades, foreign language certificate data, etc. SHAP was used to explain the model results and explain the importance of the features. In [34], student demographic information and clickstream data were used to predict at-risk students with an explainable model using Lime. The explainable model ensures that the personalized intervention should not depend on the demographic data of the students.

In [35], SHAP was used in explaining success prediction from student data such as midterm exam grade, procrastination time, correctness, the total number of logical lines in code, copy-paste information, etc. They used these features to predict whether a student passes or fails a course. Although they used programming information, the model made its predictions primarily based on the grade of the midterm exam, which was the most important feature. Courses without a first-exam grade cannot be properly assisted using their approach. Moreover, they set a hard threshold for passing and failing based on the mean grade of the course, which is not a real-world scenario as different students may follow different distributions [38]. Similarly, in [9], LIME was used to explain ML models to predict student performance from their course information, student data, and features such as clickstream and activity information (quizzes, surveys, etc.).

To the best of our knowledge, no previous study has been done on predicting student success in their final exam from their programming assignment submission information using explainable models to help instructors and students understand why someone is struggling or doing better. Therefore, this study will help in intervention with more transparency and confidence in ML model outcomes.

3. DATASET

In this study, we use a publicly available dataset [1] collected from the CodeWorkout platform. CodeWorkout [2] is an online platform that helps students practice programming in Java and allows instructors to design learning activities in their programming courses [11]. CodeWorkout logs student programming code submission information associated with different assignments. These assignments test the student's knowledge of basic programming concepts, such as data types, arrays, strings, loops, conditional statements, methods, etc.

The dataset consists of two semesters: Spring 2019 and Fall 2019. The total number of students is 772. Every semester, there are 50 programming assignments. Each assignment submission can get a score in the range of (0, 1). The number of passing test cases determines the score, and a correct submission gets a score of 1. A student can submit each assignment multiple times. The dataset consists of code submissions for each assignment and other relevant information, some of them described in Table 1.

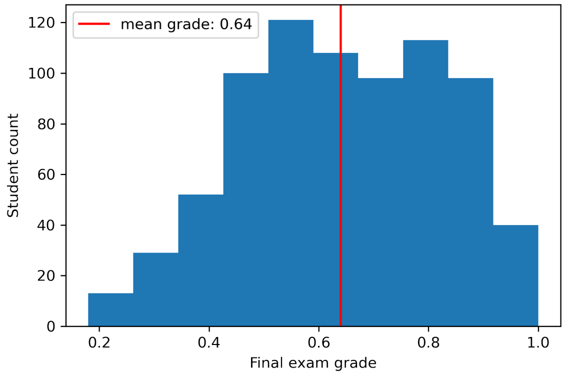


Figure 1: Distribution of the final exam grades

The CodeWorkout dataset also includes students' final exam grades, scaled between 0 and 1. The final exam grade distribution is illustrated in Figure 1. It also shows the mean final exam grade (0.64 with a standard deviation of 0.18) with a red vertical line. In this study, we try to predict the final exam grades of the students from their programming assignments in a course.

4. METHODOLOGY

To predict the final exam grades of students based on their programming submission data and explain the predictive model's predictions, we follow three steps: i) Feature engineering and extracting data-driven features from the programming submission data, ii) Developing and employing regression models to predict the final exam grades, and iii) Using SHAP to explain the model's decisions. The overall architecture of the model is illustrated in Figure 2.

4.1 Feature Engineering

We select several features associated with students' programming submissions, including total programming time spent (TimeSpent), number of unique assignments attempted (Valid), number of correct submissions (CorrectSub), number of incorrect submissions (IncorrectSub), number of uncompilable submissions (CompileError), total scores in all submissions (Scores), and total changes in codes (EditDistance). These features are described in detail below. The values of each feature are normalized to fit the range of 0-1, and a statistical description of the features is provided in Table 2.

- **TimeSpent**: It is calculated using the ServerTime of each assignment submission. The difference between the first submission for an assignment and the final submission is calculated for each assignment. The total

[1] https://pslcdatashop.web.cmu.edu/Files?datasetId=3458
[2] https://codeworkout.cs.vt.edu

Table 1: Description of student programming submission-related information in the dataset

Information	Description
SubjectID	A unique ID for every student
ToolInstances	Platform used to evaluate the code: Java 8, CodeWorkout
ServerTime	Time stamp for each submission instance
Assignment ID/ Problem ID	Unique IDs for all 50 assignments
EventType	Flag to understand if a program is compilable or not
Score	Score for each submission
CodeStateID	ID for every code submission, maps with the code of that submission
CompileMessage	Message from the compiler if there is any syntax error

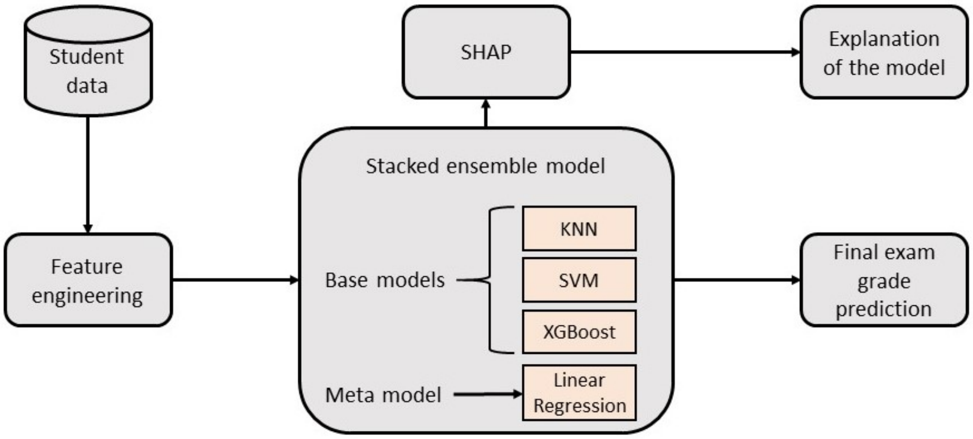


Figure 2: Architecture of the explainable model

Table 2: Statistics of the selected features

Feature	Mean (std)
TimeSpent	0.07 (0.09)
Valid	0.93 (0.11)
CorrectSub	0.41 (0.08)
IncorrectSub	0.15 (0.12)
CompileError	0.16 (0.11)
Scores	0.80 (0.13)
EditDistance	0.32 (0.16)

TimeSpent for a student is measured by adding these time differences for all attempted assignments of that student. This represents the amount of time a student has spent on solving the assignments.

- **Valid**: It counts the number of unique assignments a student attempted out of all 50 assignments. It could be a correct submission or an incorrect one.
- **CorrectSub**: It is the number of correct compilable submissions out of all the assignments. These submissions pass all the test cases and obtain a score of 1 out of 1.
- **IncorrectSub**: It is the count of incorrect submissions submitted by a student. These are compilable codes with scores less than 1 and fail some of the test cases.
- **CompileError**: It is the total number of uncompilable codes a student submits. These codes usually contain one or more syntax errors, and test cases cannot be tested on them. These submissions do not have any scores.
- **Scores**: Each assignment can have multiple incorrect and correct submissions and, thus, multiple scores. These scores are summed and normalized to have a single score for each assignment. This feature is the summation of the normalized scores of all assignments for a student.
- **EditDistance**: It is the measure of how much a student has changed the code in subsequent submissions for assignments. It is calculated using the Levenshtein algorithm. Edit distances for all assignments are summed to get an idea of a student's code change throughout the semester.

4.2 Predictor Model for Prediction

In this study, we develop a stacked ensemble regression model [50] to combine the predictive capabilities of multiple ML models. It uses a meta-learning approach to harness the powers of different models and make a final prediction. This way, the ensemble model can have better predictive power than any single predictor model individually [48]. As there are multiple predictor models, stacking uses another model that learns when to use or trust among the ensemble models.

Stacked ensemble models are different from other ensemble models, such as bagging or boosting models. Bagging is an ensemble model that combines the decision of many decision trees. Unlike bagging, stacked models are typically different in stacking (not all decision trees). In boosting, each ensemble model tries to correct the prediction of the prior models. Unlike boosting, stacking uses another ML model that learns to combine the predictions of the contributing models. In this study, bagging and boosting are used as baseline models.

Therefore, the architecture of a stacked model can be divided into two model categories:

- *Base models (Level 0):* Models that are stacked and fit on the dataset and whose predictions are combined later.
- *Meta model (Level 1):* Model that learns how to combine and trust the predictions of the base models.

This study uses KNN, SVM, and XGBoost [15] as the base models and linear regression as a metamodel to combine the predictions of the base models. While choosing the base models, diverse ML models are employed that make different assumptions regarding the prediction task. On the other hand, the meta-model is typically simpler to provide a smooth interpretation of the predictions made by the base models.

The meta-model is trained on the predictions made by the base models on hold-out data. Hold-out data is a portion of the dataset held out from the base models during training. Afterward, these hold-out data are fed to the base models to get predictions on them. These predictions from the base models on the hold-out data and the expected outputs provide the input and output pairs to train the meta-model. To train the stacked ensemble model properly, we use repeated K-fold cross-validation with 10 folds and 10 repeats.

4.3 Baseline Models

No-skill: We select the mean final exam grade as our no-skill baseline model. This naive model predicts the mean of all the student's final exam grades with no knowledge of how to make the prediction.

ML models: We also use different models to compare the performance of our stacked ensemble model. We choose the individual baseline models to see the difference in performance between our stacked ensemble model and the baseline models individually. We also use bagging and boosting to see the difference with other ensemble models. We tune the parameters of the models individually using a repeated 10-fold cross-validation approach.

- *Linear regression*
  Linear regression is a simple model which assumes a linear relationship between the inputs and outputs.
- *K-Nearest Neighbors*
  KNN stores all the available data points from the training data and predicts from k neighbors' target values based on a distance function. We select $k = 20$ and use Manhattan distance to find the neighborhood for the best result using repeated 10-fold cross-validation.
- *Support Vector Machine*
  SVM can acknowledge the presence of non-linearity in data when used in regression tasks. We set the *kernel* to rbf and the regularization parameter C to 1 for the best results.
- *Extreme Gradient Boosting*
  XGBoost is an efficient gradient-boosting-based ensemble algorithm. It outperforms other ensemble algorithms with its high efficiency and faster nature due to the parallelization of trees. We set the parameters $max\_depth = 6$, $n\_estimator = 20$, and $gamma = 1$ for the best result.
- *Bagging*
  Bagging is an ensemble model that combines the output of many decision trees. We set $n\_estimator = 10$, and $max\_features = 1$ for the best result.
- *Boosting*
  We use a Gradient Boosting regressor to represent boosting. In Boosting, each model tries to minimize the error of the prior predictor models. We set $loss$ to $squared\_error$, $learning\_rate = 0.1$, and $n\_estimators$ $= 100$ for the best result.

4.4 Explainable Artificial Intelligence (XAI) Using SHAP

ML models are black boxes in nature. Many applications of ML require explanations of the decisions made by the models depending on the stakeholders. Explanations of the decisions are vital parts of working with populations like students and learners. Such interpretable and explainable models can provide insights into the effectiveness of students' problem-solving strategies and enable instructors and advanced learning technologies to provide students with effective formative feedback. These can also help in gaining the trust of students and instructors by understanding their reasonings behind such decisions.

SHapley Additive exPlanation (SHAP) [27] is an adaptive algorithm based on the Game theory [41]. In this framework, the variability of predictions is split into the features used in the prediction model. Therefore, the contribution and importance of each feature behind the predictive model (global) and individual predictions (local) can be measured in a model-agnostic way [18].

SHAP calculates Shapley values for each feature for each instance. These values determine the presence of each covariate in the model predictions as a linear combination of each predictor variable. To calculate the positive or negative effect of each feature on the predictions, the algorithm examines the change in each prediction when a feature $i \in F$ is withheld, where F is the set of all features [27]. Thus, the feature importance of a feature i for a model f is calculated by the evaluation of the marginal contribution $\Phi_i \in \mathbb{R}$ for all the subsets $S \subseteq F$. According to [26], to satisfy local accuracy, consistency, and missingness properties, Φ_i (Shapley values) defined as:

$$\Phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)]$$

Where, Φ_i is the marginal contribution of feature i on the model's output $f_{S\cup\{i\}}(x_{S\cup\{i\}})$.

By the additive property, SHAP approximates each prediction $f(x)$ of the model with the help of $f'(y')$ which is a linear combination of all binary variables $y' \in \{0,1\}^N$ (N is the maximum size for the simplified feature vectors) and the marginal contributions of each feature Φ_i in such a way that the sum of all the feature contributions should match the output of f for a simplified input y':

$$f(x) = f'(y') = \Phi_0 + \sum_{i=1}^{N} \Phi_i \cdot y'_i$$

Where, Φ_0 is the expected prediction without any prior information, in our case, the average final exam grades of students. In a nutshell, Shapley values approximate a model's predictions locally for a given variable x (local accuracy). It tries to ensure that the contribution of a variable is zero when the variable is zero (missingness). If the contribution of a variable is higher in the model's prediction, then the Shapley value for that variable should also be higher (consistency).

5. RESULTS

In this section, we evaluate the performance of our proposed stacked ensemble model in students' final exam grade prediction. Later, we interpret the black-box model by analyzing the importance and impact direction of each of the features using SHAP. We further analyze the influence of the most important features on the final exam grades to categorize students into different profiles based on their performances.

5.1 Evaluation

We compare our results with the base models, the meta-model, and other ensemble models individually. Therefore, we experiment with Linear regression, KNN, SVM, XGBoost, Bagging, and Boosting in the same task. We also use a no-skill model, which predicts the mean final exam grade. This acts as a naive baseline model without prior knowledge of the features. All the models are evaluated using a 10-fold cross-validation approach with ten repeats to get a stable result.

To compare the performances of these models, we measure the root-mean-square error (RMSE) of the predicted final exam grades with respect to the actual final exam grades. Since RMSE follows the same range (0-1) as our final exam grades, it can provide insights into how far the predicted values are from the actual ones. Moreover, it penalizes large errors. This makes it a suitable metric to evaluate the model performances since models with a consistent and stable accuracy level are more useful than models with more errors, and RMSE gives relatively high weight to large errors [51]. We also use R^2 along with RMSE as the evaluation metric. The coefficient of determination (R^2) shows how much of the variation in the dependent variable is accounted for by the independent variables in a regression model.

Table 3: Performance comparison of different models

Model	RMSE	R^2
no-skill	0.247 (0.020)	-0.01 (0.018)
Linear	0.185 (0.004)	0.38 (0.08)
KNN	0.173 (0.004)	0.37 (0.10)
SVM	0.159 (0.003)	0.51 (0.10)
XGBoost	0.170 (0.005)	0.39 (0.09)
Bagging	0.166 (0.004)	0.44 (0.09)
Boosting	0.161 (0.003)	0.47 (0.08)
Stacked ensemble	0.151 (0.003)	0.55 (0.07)

Table 3 depicts the performances of the regression models based on RMSE values and R^2 scores. These values are calculated by taking the average of the repeated cross-validation results. The standard deviation of each model is also calculated and shown in parentheses with the average RMSE and R^2. We can see that all the regression models outperform the naive baseline model with no skill. Our stacked ensemble regression model outperforms all other models with an RMSE of 0.151 and an R^2 score of 0.55. We further investigate the performance of our model statistically to see if the model's performance is significantly different from other models. We use the Wilcoxon-signed rank test with a significance level of 0.05. The null hypothesis is that the performance of our model is the same as any other model. The null hypothesis is rejected for all the baseline models (p-value<0.05). Additionally, we test our model's performance using half of the assignments (first 25 out of 50, ordered with assignment ID). The RMSE value is 0.18 (0.005), and the R^2 score is 0.41 for our model, which is also higher than other models while using only half of the assignments of the course. These results prove that our model shows statistically significant improvement over the performances of other models.

5.2 Unfolding the Blackbox Model

To better understand the underlying mechanism behind the stacked ensemble model's predictions, we calculate the Shapley values, values that determine the importance and impact direction of each feature, using the SHAP algorithm. Using SHAP, we can get the interpretation at an individual level for a student as well as a global level for all students. It enables us to understand the model predictions in a transparent way.

5.2.1 Individual Level Explanation

At first, we look at an individual student's final exam grade prediction made by the model. Figure 3 shows a force plot for an individual student whose actual final exam grade is 0.61. $f(x)$ is the model's prediction which is 0.59. The base value is 0.64, which is the mean final exam grade. This is the prediction of the no-skill model if there is no prior knowledge about the features. The plot also shows the most important feature names and their corresponding values for this prediction. The red-colored features pushed the predicted final grade higher, and the blue-colored features pushed the grade lower. The longer the arrow is, the larger the impact of that feature on the decision. Low EditDistance, CompilerError, and high Valid helped the predicted grade to be higher, whereas high Scores and TimeSpent pushed the grade to be lower. We plot the relative importance of features in Figure 4 using SHAP to understand the force plot

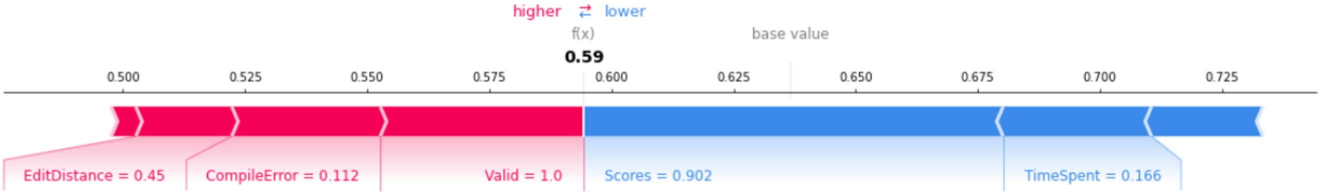


Figure 3: Force plot for an individual student

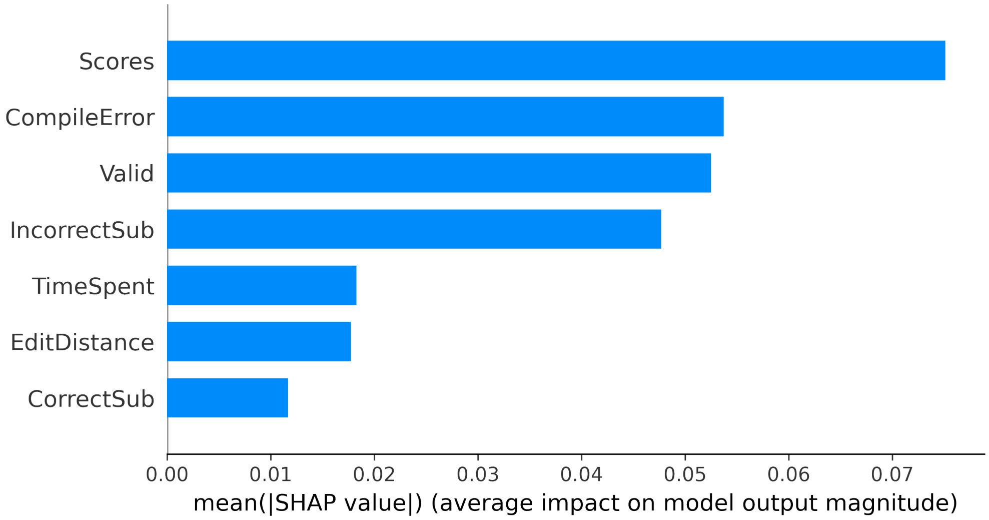


Figure 4: Relative importance of features

more clearly and to comprehend the relative importance of features. The X-axis represents the relative importance of the features on the model's predictions. We can see that Scores is the most important feature, whereas CorrectSub has the least importance.

5.2.2 Global Level Explanation

We plot the summary of all the features at a global level for all students to understand the relationship between feature values and predicted values in Figure 5. In the summary plot, the features are ranked by importance. Each point represents the Shapley value for each feature regarding prediction for a single data point. Overlapping points are jittered around the Y-axis to get an idea of the distribution of the Shapley values. Red represents a high value for that feature, and blue represents a low value. The summary plot shows that students with high CompileError, low Valid, high TimeSpent, low EditDistance, and low CorrectSub have negative Shapley values, which correspond to a higher probability of performing poorly in the final exam. Therefore, students with a relatively high number of compiler errors, low number of attempted assignments, high amount of time spent on submitting the assignments, low changes or edits in subsequent submissions, and low number of correct assignments have negative Shapley values, which correspond to a lower final exam grade.

On the other hand, the feature impact of Scores and IncorrectSub on students' final grades are demonstrated as counterintuitive results based on the summary plot. We can see that some students with lower scores tend to have higher grades in the final exam, while some students with higher scores do not do well in the exam. Similarly, some students with a higher number of incorrect submissions do well in the final exam, while some students with a lower number of incorrect submissions achieve poor grades in the exam. We hypothesize that this observation can be explained by looking more closely at students' programming behavior, including their average edit distance in each submission. Other prior works have used the edit distance to group students based on their problem-solving behavior and identify effective problem-solving patterns based on each group's performance [4, 3]. Thus, we hypothesize that the interaction between scores, incorrect submissions, and edit distance can be deterministic of students' learning. To investigate our hypothesis, we discretized the values for scores and the number of incorrect submissions features into low and high values using Gaussian Mixture Modeling (GMM) [38]. GMM can be used for clustering where probabilistically, each data point is assumed to be generated from a mixture of a finite number of Gaussian distributions where the parameters are unknown. It uses Expectation-Maximization (EM) algorithm to determine these parameters. Each Gaussian distribution is specified by its mean and covariance.

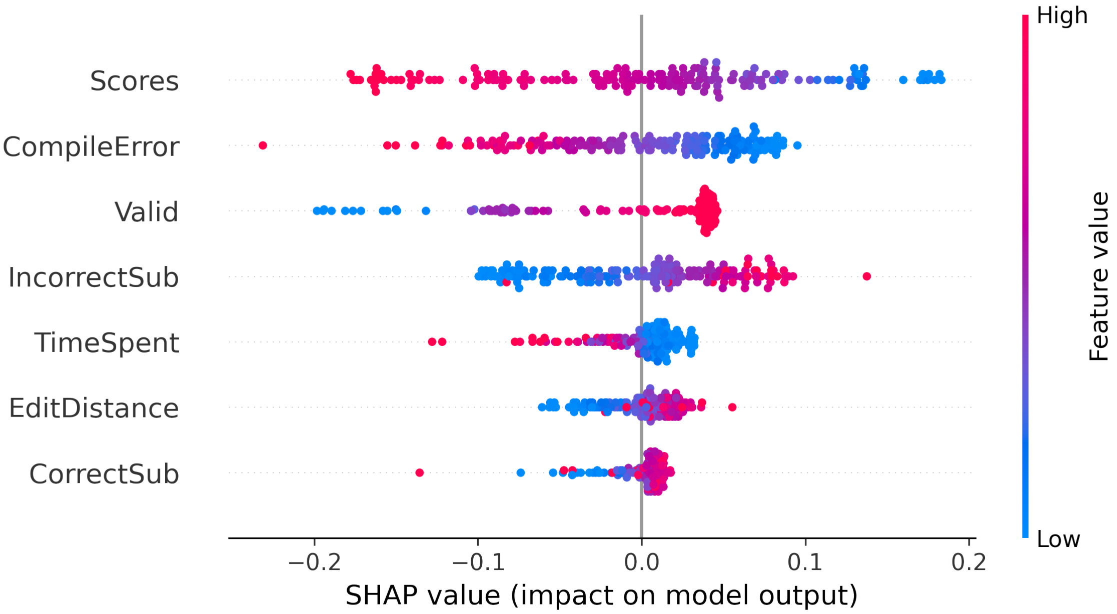


Figure 5: Summary plot showing feature importance with their impacts

We use Gaussian Mixture Modeling to divide each feature distribution into two components: "component low" and "component high". Students belonging to "component low" has a relatively lower value, and "component high" has relatively higher values for that individual feature. Figure 6 shows the components of the feature Scores where "component low" has a mean Scores value of 0.71 and "component high" has a mean Scores value of 0.87. Similarly, figure 7 shows the components of the feature IncorrectSub, where "component low" has a mean IncorrectSub value of 0.08, and "component high" has a mean IncorrectSub value of 0.26. From the components of each feature obtained from Gaussian Mixture Modeling, we get the students of "component low" for each feature whose probability of being assigned to "component low" is higher than that of being assigned to "component high". Similarly, we get the students who belong to "component high" for each feature. After investigating the interactions between these two features, students' final exam grades, and also taking the impact of edit distance into account, we identified three main student profiles and named them with the help of expert CS educators, based on possible values for the number of incorrect submissions and their average programming assignment scores [7, 24, 23]. These profiles are demonstrated in Table 4, along with each profile's average final grades.

5.3 Student Profiling

As discussed previously, we further analyze the explanations of the model's predictions to profile students based on their learning outcomes and strategies.

5.3.1 *Expert or Cheating Students*

Students who have a high score and a low number of incorrect submissions on average have a mean final exam grade of 0.58. This grade is 0.06 lower than the overall mean grade (0.64). We hypothesize that students who submit a low number of incorrect submissions with a high score on average are either experts or cheaters cheating from expert students and not learning enough, and thus, we expect to see a noticeable difference between the average final grade for these two sets of students.

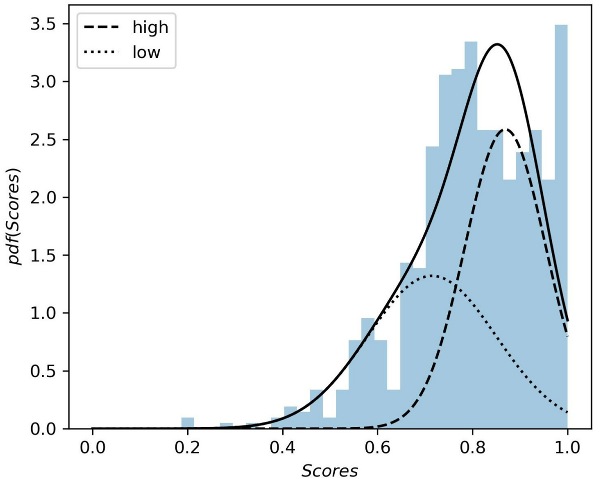


Figure 6: Components of feature: Scores

To test this hypothesis, we divide these students into expert and cheating profiles based on their final exam grades and check the mean grades of these two profiles to determine whether there is a significant difference between them. The cheating group has a mean final exam grade of 0.48, and the expert group has a mean final exam grade of 0.80.

Table 4: Student profiles based on Scores and IncorrectSub values

Scores	IncorrectSub	Student Profile	Final exam mean (std)
low	high	Learning	0.72 (0.14)
low	low	Struggling	0.60 (0.18)
high	low	Expert Cheating	0.80 (0.08) 0.48 (0.10)
high	high	Outlier	0.68 (0.19)

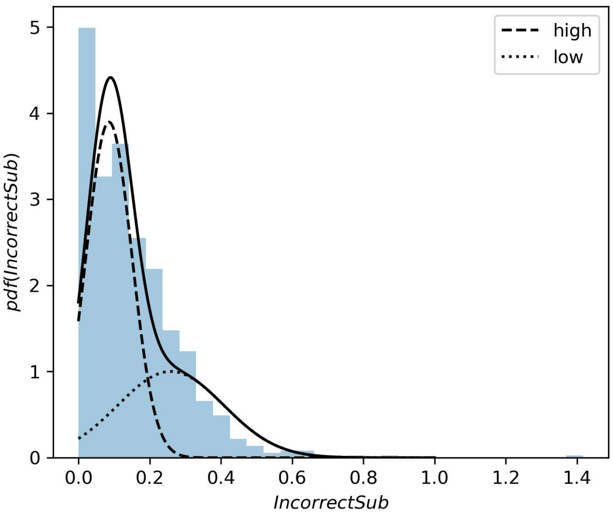


Figure 7: Components of feature: IncorrectSub

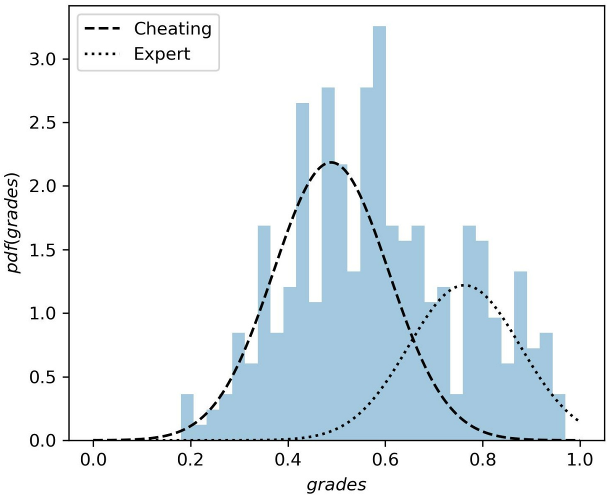


Figure 8: Components of cheating and expert students

Moreover, the grades of these two profiles are statistically different with a p-value of less than 0.05. We further verify our hypothesis using Gaussian Mixture Modeling and dividing the final exam grade distribution of this profile into two components [38] as illustrated in Figure 8. The component with a high mean grade (0.79) represents the expert group, and the component with a low mean grade (0.49) represents the cheating group. This is a clear indication that there is a significant difference in the competency of both groups.

5.3.2 Learning Students

The second profile of students we investigate are students with a high number of incorrect submissions and a low score on average. This group has a mean final exam grade of 0.72, which is 0.08 higher than the total average mean grade. About 84% of these students have high edit distance on average. This suggests that students without a solid background knowledge learn through trial and error by attempting different solutions multiple times.

5.3.3 Struggling Students

Students who have a low number of incorrect submissions and lower scores on average are identified as struggling students. This group has a mean final exam grade of 0.6, which is 0.04 points lower than the mean grade of all students. About 89% of these students have low edit distance on average. The mean EditDistance for this group is 0.2 which is 0.12 points lower than the average edit distance for all students. They also have 0.3 points mean for the number of correct submissions which is 0.11 points lower than the average number of correct submissions.

5.3.4 Outlier

The last group of students is students who have a high number of incorrect submissions with a high score on average. This group of students constitutes as low as 10% of the total dataset and, thus, is not investigated further and is not included in any particular profiles.

These results suggest that while expert students can get the desired outcome through a few high-quality attempts, students with moderate levels of knowledge and expertise aim for the desired results through multiple incorrect submissions, attempting new solutions for each submission. On the lower end of the spectrum, struggling students would not put any effort into engaging with the activities as demonstrated through a low number of submissions with a low score on average. These analytical results explain why features Scores and IncorrectSub have an atypical effect on the model's predictions.

On the whole, our stacked ensemble model can effectively predict students' final exam grades using their programming assignment information. The results from incorporating the SHAP model can shed light on students' problem-solving strategies and the connection between those strategies and students' learning outcomes. Utilizing an explainable model to perform prediction of students' performance can help instructors and advanced learning technologies make informed decisions about effective interventions based on students' progress and problem-solving patterns in a timely manner.

6. DISCUSSION

Introductory programming classes are growing rapidly while being one of the most challenging subjects for students. Thus, it is important to build automated approaches that enable instructors and educators to provide students with timely pedagogical support. We need to design generalizable and interpretable prediction models that can predict students' performance while analyzing their problem-solving behavior. However, predicting student performance in an introductory programming course, such as predicting the final exam grades solely based on programming data is a challenging task, given that the nature of the final exam differs from hands-on programming assignments. Prior research has used conventional classroom data such as tests, exams, and multiple choice grades to predict students' final exam grades. On the other hand, including programming features have been shown to improve the prediction results in computer science courses [35]. While the exam grades can certainly improve the results, it is not always accessible to the CSEDM models due to limitations in data collection, such as data privacy. Furthermore, different introductory programming classes might have different outlines and grading mechanisms, while almost all of them include programming assignments. Thus, our model can be generalized to any introductory programming course regardless of its outline since it merely relies on programming assignment data.

Integrating our approach in a classroom can offer valuable benefits to both students and instructors. First, the explainable stacked ensemble model developed in this study can help identify struggling students by predicting their final exam grades based on their programming assignment data. By identifying struggling students, instructors can offer targeted interventions and support to help them improve their learning outcomes. Second, the explanations of the model's predictions using the SHAP algorithm can help students and instructors understand the model's decision-making process. This understanding can help build trust in the model's predictions. Additionally, the study's interpretation of the SHAP results as profiles that group students based on their problem-solving strategy patterns can provide valuable insights into students' problem-solving behavior and learning outcomes [13, 23]. This information can help instructors develop personalized teaching strategies that cater to each group's unique needs, thus enabling more effective interventions and support [8, 31, 42].

There are a few limitations in this work. First, our dataset did not allow for early prediction of students' performance since students could have attempted the assignments in an arbitrary order at any point in time. However, we trained our model with a subset of assignments to test the generalizability of the model in courses where fewer assignments are available. Our model outperformed other baselines significantly with fewer numbers of assignments. Additionally, the dataset used in this study has potential plagiarism issues. Plagiarism affects the performance of our model because programming submission information and problem-solving pattern do not convey the actual information about the cheating students' learning. Moreover, the dataset lacks sufficient contextual information related to the course and the CodeWorkout implementation. In particular, there is no information on dropped-out students and students who missed the final exam. The final exam grades of these students are stated as zero (less than 1% of the dataset), which might affect the performance of a predictive model.

7. CONCLUSION

In this study, we extracted important data-driven features from students' programming submissions that can be representative of students' problem-solving behavior and utilized them to predict students' performance. Furthermore, we developed an explainable stacked ensemble model that can predict students' final exam grades from their programming assignment information. Our model could significantly outperform baseline models, including Linear regression, KNN, SVM, XGBoost, Bagging, and Boosting. The predictions made by our model were explained using the SHAP algorithm that shows the importance and direction of impacts for each feature with regard to the predictions. We have provided explanations of the decisions made by the model at two levels: explanations of the decision for a student at an individual level and explanations of the overall predictions at a global level. This explanation can help students and instructors to understand the model's predictions and make it trustworthy. We used a combination of descriptive statistical analysis and mixture models to interpret the SHAP results as profiles that group students based on their problem-solving strategy patterns. This enables us to gain insights into students' problem-solving behavior and connection to their learning outcomes.

In the future, we intend to utilize our model for early prediction by training it on a dataset where students' attempts at assignments follow a specified order. This will also facilitate analyzing student profiles, programming-solving behaviors, and patterns at different stages of the course time. Moreover, investigating students' problem-solving strategies for individual assignments with different difficulties might help us to understand students' struggles associated with different concepts represented by each assignment. In this study, student profiling was used by discretizing the students into two components (low and high) based on each feature value to analyze the SHAP values where feature impact on the predictions was not straightforward and counterintuitive. Nonetheless, if we consider more than two components for each feature for a more complex student body, more student profiles might emerge in the interpretation process based on the feature interactions. We intend to explore more complex situations and analyze explanations obtained from SHAP with more granular student profiles in the future. Furthermore, we intend to conduct in-depth studies to detect plagiarism and cheating in students' programming codes. This includes strategies for similarity analysis and anomaly detection. For instance, we can assess the similarity between two codes through program embedding approaches where the structural information of each program is captured through vectors. Moreover, we can analyze students' normalized submission rate distributions to identify odd patterns for a particular assignment to gain insights into the likelihood of students committing plagiarism over the course of time.

8. REFERENCES

[1] A. Ahadi, R. Lister, H. Haapala, and A. Vihavainen. Exploring machine learning methods to automatically identify students in need of assistance. In *Proceedings of the 11th Annual International Conference on International Computing Education Research*, pages 121–130, 2015.

[2] I. Ahmed, G. Jeon, and F. Piccialli. From artificial intelligence to explainable artificial intelligence in industry 4.0: a survey on what, how, and where. *IEEE Transactions on Industrial Informatics*, 18(8):5031–5042, 2022.

[3] B. Akram, W. Min, E. Wiebe, B. Mott, K. E. Boyer, and J. Lester. Improving stealth assessment in game-based learning with lstm-based analytics. In *Proceedings of the International Conference on Educational Data Mining*, pages 208–218, 2018.

[4] B. Akram, W. Min, E. Wiebe, B. Mott, K. E. Boyer, and J. Lester. Assessing middle school students' computational thinking through programming trajectory analysis. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, pages 1269–1269, 2019.

[5] N. Alam, H. Acosta, K. Gao, and B. Mostafavi. Early prediction of student performance in a programming class using prior code submissions and metadata. In *Proceedings of the 6th Educational Data Mining in Computer Science Education (CSEDM) Workshop*, pages –, 2022.

[6] M. Baranyi, M. Nagy, and R. Molontay. Interpretable deep learning for university dropout prediction. In *Proceedings of the 21st Annual Conference on Information Technology Education*, pages 13–19, 2020.

[7] M. S. Boroujeni and P. Dillenbourg. Discovery and temporal analysis of latent study patterns in mooc interaction sequences. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, pages 206–215, 2018.

[8] A. Boubekki, S. Jain, and U. Brefeld. Mining user trajectories in electronic text books. In *the 11th International Conference on Educational Data Mining (EDM)*, pages 147–156, 2018.

[9] H.-C. Chen, E. Prasetyo, S.-S. Tseng, K. T. Putra, S. S. Kusumawardani, and C.-E. Weng. Week-wise student performance early prediction in virtual learning environment using a deep explainable artificial intelligence. *Applied Sciences*, 12(4):1885, 2022.

[10] E. B. Costa, B. Fonseca, M. A. Santana, F. F. de Araújo, and J. Rego. Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses. *Computers in Human Behavior*, 73:247–256, 2017.

[11] S. H. Edwards and K. P. Murali. Codeworkout: short programming exercises with built-in data collection. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*, pages 188–193, 2017.

[12] O. Embarak. Explainable artificial intelligence for services exchange in smart cities. *Explainable Artificial Intelligence for Smart Cities*, pages 13–30, 2021.

[13] N. Gitinabard, S. Heckman, T. Barnes, and C. F. Lynch. What will you do next? a sequence analysis on the student transitions between online platforms in blended courses. In *the 12th International Conference on Educational Data Mining (EDM)*, pages 59–68, 2019.

[14] M. Hoq, P. Brusilovsky, and B. Akram. SANN: A subtree-based attention neural network model for student success prediction through source code analysis. In *6th Educational Data Mining in Computer Science Education (CSEDM) Workshop*, pages –, 2022.

[15] M. Hoq, M. N. Uddin, and S.-B. Park. Vocal feature extraction-based artificial intelligent model for parkinson's disease detection. *Diagnostics*, 11(6):1076, 2021.

[16] P. Ihantola, A. Vihavainen, A. Ahadi, M. Butler, J. Börstler, S. H. Edwards, E. Isohanni, A. Korhonen, A. Petersen, K. Rivers, et al. Educational data mining and learning analytics in programming: Literature review and case studies. In *2015 ITiCSE on Working Group Reports*, pages 41–63, 2015.

[17] M. Jamjoom, E. Alabdulkreem, M. Hadjouni, F. Karim, and M. Qarh. Early prediction for at-risk students in an introductory programming course based on student self-efficacy. *Informatica*, 45(6), 2021.

[18] A. Joseph. Shapley regressions: A framework for statistical inference on machine learning models. *arXiv preprint arXiv:1903.04209*, 2019.

[19] M. S. Kamal, A. Northcote, L. Chowdhury, N. Dey, R. G. Crespo, and E. Herrera-Viedma. Alzheimer's patient analysis using image and gene expression data and explainable-ai to present associated genes. *IEEE Transactions on Instrumentation and Measurement*, 70:1–7, 2021.

[20] H. Karimi, T. Derr, J. Huang, and J. Tang. Online academic course performance prediction using relational graph convolutional neural network. In *Proceedings of the 13th International Conference on Educational Data Mining (EDM)*, pages 444–450, 2020.

[21] I. Khan, A. Al Sadiri, A. R. Ahmad, and N. Jabeur. Tracking student performance in introductory programming by means of machine learning. In *2019 4th MEC International Conference on Big Data and Smart City (ICBDSC)*, pages 1–6. IEEE, 2019.

[22] E. J. Lauría, J. D. Baron, M. Devireddy, V. Sundararaju, and S. M. Jayaprakash. Mining academic data to improve college student retention: An open source perspective. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, pages 139–142, 2012.

[23] E. Loginova and D. F. Benoit. Embedding navigation patterns for student performance prediction. In *14th International Conference on Educational Data Mining (EDM)*, pages 391–399, 2021.

[24] S. Lorenzen, N. Hjuler, and S. Alstrup. Tracking behavioral patterns among students in an online educational system. In *Proceedings of the 11th International Conference on Educational Data Mining (EDM)*, pages 280–285, 2018.

[25] Y. Lu, D. Wang, Q. Meng, and P. Chen. Towards interpretable deep learning models for knowledge

tracing. In *Proceedings of the 21st International Conference on Artificial Intelligence in Education (AIED)*, pages 185–190. Springer, 2020.

[26] S. M. Lundberg, G. G. Erion, and S.-I. Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018.

[27] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st international Conference on Neural Information Processing Systems*, volume 30, pages 4768–4777, 2017.

[28] Y. Mao. One minute is enough: Early prediction of student success and event-level difficulty during novice programming tasks. In *Proceedings of the 12th International Conference on Educational Data Mining (EDM)*, pages 119–128, 2019.

[29] Y. Mao, F. Khoshnevisan, T. Price, T. Barnes, and M. Chi. Cross-lingual adversarial domain adaptation for novice programming. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7682–7690, 2022.

[30] J. Marsden, S. Yoder, and B. Akram. Predicting Student Performance with Control-flow Graph Embeddings. In *6th Educational Data Mining in Computer Science Education (CSEDM) Workshop*, pages –, 2022.

[31] K. Mouri, A. Shimada, C. Yin, and K. Kaneko. Discovering hidden browsing patterns using non-negative matrix factorization. In *the 11th International Conference on Educational Data Mining (EDM)*, pages 568–571, 2018.

[32] T. Mu, A. Jetten, and E. Brunskill. Towards suggesting actionable interventions for wheel-spinning students. In *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, pages 183–193, 2020.

[33] S. M. Muddamsetty, M. N. Jahromi, and T. B. Moeslund. Expert level evaluations for explainable ai (xai) methods in the medical domain. In *International Conference on Pattern Recognition*, pages 35–46. Springer, 2021.

[34] B. Pei and W. Xing. An interpretable pipeline for identifying at-risk students. *Journal of Educational Computing Research*, 60(2):380–405, 2022.

[35] F. D. Pereira, S. C. Fonseca, E. H. Oliveira, A. I. Cristea, H. Bellhäuser, L. Rodrigues, D. B. Oliveira, S. Isotani, and L. S. Carvalho. Explaining individual and collective programming students' behavior by interpreting a black-box predictive model. *IEEE Access*, 9:117097–117119, 2021.

[36] F. D. Pereira, E. Oliveira, A. Cristea, D. Fernandes, L. Silva, G. Aguiar, A. Alamri, and M. Alshehri. Early dropout prediction for programming courses supported by online judges. In *International Conference on Artificial Intelligence in Education*, pages 67–72. Springer, 2019.

[37] K. Quille and S. Bergin. Cs1: how will they do? how can we help? a decade of research and practice. *Computer Science Education*, 29(2-3):254–282, 2019.

[38] M. Sahami and C. Piech. As cs enrollments grow, are we attracting weaker students? In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pages 54–59, 2016.

[39] O. Serradilla, E. Zugasti, C. Cernuda, A. Aranburu, J. R. de Okariz, and U. Zurutuza. Interpreting remaining useful life estimations combining explainable artificial intelligence and domain knowledge in industrial machinery. In *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE, 2020.

[40] A. M. Shahiri, W. Husain, et al. A review on predicting student's performance using data mining techniques. *Procedia Computer Science*, 72:414–422, 2015.

[41] L. S. Shapley. Quota solutions of n-person games. Technical report, RAND CORP SANTA MONICA CA, 1952.

[42] A. Sheshadri, N. Gitinabard, C. F. Lynch, T. Barnes, and S. Heckman. Predicting student performance based on online study habits: A study of blended courses. In *the 11th International Conference on Educational Data Mining (EDM)*, pages 87–96, 2018.

[43] Y. Shi, R. Schmucker, M. Chi, T. Barnes, and T. Price. KC-Finder: Automated knowledge component discovery for programming problems. In *Proceedings of the 16th International Conference on Educational Data Mining (EDM)*, pages –, 2023.

[44] A. Singh, S. Sengupta, M. A. Rasheed, V. Jayakumar, and V. Lakshminarayanan. Uncertainty aware and explainable diagnosis of retinal disease. In *Medical Imaging 2021: Imaging Informatics for Healthcare, Research, and Applications*, volume 11601, pages 116–125. SPIE, 2021.

[45] Q. Sun, J. Wu, and K. Liu. Toward understanding students' learning performance in an object-oriented programming course: The perspective of program quality. *IEEE Access*, 8:37505–37517, 2020.

[46] M. Sweeney, J. Lester, H. Rangwala, A. Johri, et al. Next-term student performance prediction: A recommender systems approach. *Journal of Educational Data Mining*, 8(1):22–51, 2016.

[47] D. Thakker, B. K. Mishra, A. Abdullatif, S. Mazumdar, and S. Simpson. Explainable artificial intelligence for developing smart cities solutions. *Smart Cities*, 3(4):1353–1382, 2020.

[48] K. M. Ting and I. H. Witten. Stacked generalization: when does it work? In *Poceedings of the 15th Joint International Conference on Artificial Intelligence*, pages 866–871, 1997.

[49] C. Watson and F. W. Li. Failure rates in introductory programming revisited. In *Proceedings of the 2014 conference on Innovation & technology in computer science education*, pages 39–44, 2014.

[50] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

[51] S. Yoder, M. Hoq, P. Brusilovsky, and B. Akram. Exploring sequential code embeddings for predicting student success in an introductory programming course. In *6th Educational Data Mining in Computer Science Education (CSEDM) Workshop*, pages –, 2022.

[52] M. Yudelson, R. Hosseini, A. Vihavainen, and P. Brusilovsky. Investigating automated student modeling in a java MOOC. In *Proceedings of the 7th International Conference on Educational Data Mining (EDM)*, pages 261–264, 2014.

Is Your Model "MADD"? A Novel Metric to Evaluate Algorithmic Fairness for Predictive Student Models

Mélina Verger
Sorbonne Université, CNRS,
LIP6, F-75005 Paris, France
melina.verger@lip6.fr

Sébastien Lallé
Sorbonne Université, CNRS,
LIP6, F-75005 Paris, France
sebastien.lalle@lip6.fr

François Bouchet
Sorbonne Université, CNRS,
LIP6, F-75005 Paris, France
francois.bouchet@lip6.fr

Vanda Luengo
Sorbonne Université, CNRS,
LIP6, F-75005 Paris, France
vanda.luengo@lip6.fr

ABSTRACT

Predictive student models are increasingly used in learning environments due to their ability to enhance educational outcomes and support stakeholders in making informed decisions. However, predictive models can be biased and produce unfair outcomes, leading to potential discrimination against some students and possible harmful long-term implications. This has prompted research on fairness metrics meant to capture and quantify such biases. Nonetheless, so far, existing fairness metrics used in education are predictive performance-oriented, focusing on assessing biased outcomes across groups of students, without considering the behaviors of the models nor the severity of the biases in the outcomes. Therefore, we propose a novel metric, the Model Absolute Density Distance (MADD), to analyze models' discriminatory behaviors independently from their predictive performance. We also provide a complementary visualization-based analysis to enable fine-grained human assessment of how the models discriminate between groups of students. We evaluate our approach on the common task of predicting student success in online courses, using several common predictive classification models on an open educational dataset. We also compare our metric to the only predictive performance-oriented fairness metric developed in education, ABROCA. Results on this dataset show that: (1) fair predictive performance does not guarantee fair models' behaviors and thus fair outcomes, (2) there is no direct relationship between data bias and predictive performance bias nor discriminatory behaviors bias, and (3) trained on the same data, models exhibit different discriminatory behaviors, according to different sensitive features too. We thus recommend using the MADD on models that show satisfying predictive performance, to gain a finer-grained understanding on how they behave and regarding who and to refine models selection and their usage. Altogether, this work contributes to advancing the research on fair student models in education. Source code and data are in open access at `https://github.com/melinaverger/MADD`.

Keywords

fairness metric, classification, student modeling, models' behaviors, sensitive features

1. INTRODUCTION

Over the past decade, extensive research has focused on predictive student modeling for educational applications. The systematic literature review of Hellas et al. [15] has identified no less than 357 relevant papers on the matter published between 2010 and mid-2018. One of the most popular modeling technique in these works are machine learning (ML) classifiers, as many important predictive tasks in education can be framed as binary classification problems, e.g. to predict dropout, course completion, university admission, scholarship awarding. These classification models have thus gained widespread adoption, and the multiple stakeholders involved in education have recognized their potential to improve student learning outcomes and experience [29, 16].

However, in recent years, there have been concerns about the fairness of the models (also called *algorithmic fairness* [3]) used in education [3, 20, 11, 33]. This stems from a more general trend of research in ML and Artificial Intelligence (AI), where a large body of research has shown that classifiers, and AI models in general, can produce biased and unfair outcomes, e.g. [27, 5, 4, 23, 10]. This has led to increased public awareness about the potential harms of AI predictive models and the enforcement of stricter regulations[1]. In education too, recent studies have found that classification-based student models can be biased against certain groups of students, which could in turn significantly hinder their learning experience and academic achievements [3, 20, 33, 14, 17, 25].

M. Verger, S. Lallé, F. Bouchet, and V. Luengo. Is your model "madd"? a novel metric to evaluate algorithmic fairness for predictive student models. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 91–102, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115786

[1] e.g. General Data Protection Regulation (2016) at European level, California Consumer Privacy Act (2018) at the United-States level, Principles on Artificial Intelligence (2019) from OECD (Organization for Economic Cooperation and Development) at the international level, and more specifically the upcoming European AI Act [32].

To unveil, measure and mitigate algorithmic unfairness, recent literature in AI has seen a proliferation of *fairness metrics* [35, 7]. Although many types of metrics exist (see Section 2), some of them require extensive prior knowledge and in practice the most common fairness metrics used in AI are statistical [35]. Statistical metrics aim at quantifying the differences in performance of a set of classification models across different groups of interest, with the assumption that fair classifiers should achieve similar performance across groups [7]. This is especially meaningful when some of the groups are known to be vulnerable to unfair model predictions. For instance, students with disability might be unfairly classified as at-risk of dropping out of an online course because the features used to train the classifiers did not capture well the different way they engage with the learning material [13] – when they can interact at all, since many K-12 material or educational technologies remain inaccessible [31, 6]. However, the pitfall of the existing statistical metrics is that they are all *predictive performance-oriented*, meaning that they solely consider the predictive performance of the classification model across predefined groups, disregarding that two classifiers with equal predictive performance can exhibit very different, and possibly unfair, behaviors. In particular, a classifier could produce similar error rates across two groups, but the actual errors made could be substantially more harmful to one of the group than the other.

In this paper, we thus propose a new statistical metric, the *Model Absolute Density Distance* (MADD), to analyze a model's discriminatory behaviors independently from its predictive performance. We also propose a complementary visualization-based analysis, which allows to inspect and qualify the models' discriminatory behaviors uncovered by the MADD. Altogether, this makes it possible to not only quantify, but also understand in a fine-grained way whether and how a given classifier may behave differently between the groups. As a case study, we apply our approach on the common task of predicting student success to a course, on open data for the sake of replication and on four common predictive classification models for the sake of generalization. We also compare our metric to ABROCA (*Absolute Between-ROC Area*), the only predictive performance-oriented metric developed in education [14] to the best of our knowledge. This case study shows that the MADD can successfully capture fine-grained models' discriminatory behaviors.

The remainder of this paper is organized as follows. Section 2 reports on related work on fairness metrics and their usage in education. Section 3 presents the MADD metric and the visualization-based analysis we propose to inspect and characterize models' discriminatory behaviors. Section 4 describes the experimental setup with which we applied our proposed approach in order to demonstrate its benefits. Section 5 presents our results and our comparison with ABROCA. In Section 6, we discuss more generally what our approach allows to unveil, the strengths and limitations it currently has as well as some practical guidelines, before concluding in Section 7 with future work.

2. RELATED WORK

Several fairness metrics have been proposed in AI for classification models. These metrics mostly fall into three categories: counterfactual (or causality-based), similarity-based (or individual), and statistical (or group) [35]. The first two categories, counterfactual and similarity-based, are seldom used in practice because they require extensive prior knowledge. More precisely, counterfactual metrics require building a directed acyclic causal graph with the nodes representing the features of an applicant and the edges representing relationships between the features [35]. Generating such a causal graph is typically not feasible without extensive studies to formally identify these relations. Similarity-based metrics require defining *a priori* a distance metric to measure how "similar" two individuals are, as well as to know from which value the models' results are considered "dissimilar" enough for these two individuals to be pointed out as unfairness. In contrast, statistical metrics, the category into which MADD falls, are easier to implement and more popular, as they solely require to identify *a priori* the groups of persons who might suffer from unfair classifications. As noted in the introduction, these metrics have so far sought to quantify differences in classification performance across the groups, and thus can be considered *predictive performance-oriented* only. However, a classifier that has similar error rates across two groups might actually produce errors that are harmless to a group but very harmful to the other, an aspect that is not quantified by existing statistical metrics. In this paper, we focus on the new MADD metric meant to assess unfair behaviors of classifiers, independently from their predictive performance. We recommend using it as a complement to a predictive performance analysis, rather than using predictive performance-oriented fairness metrics only, in order to gain a more refined and comprehensive understanding of the classifiers fairness.

In education, fairness studies are more recent and sparse (see overview in [3, 20]), and only a handful of them have focused on the fairness of classification models used in this context [14, 17, 18, 30, 25]. In Gardner et al. [14], the authors propose a new predictive performance-oriented fairness metric based on the comparison of the Areas Under the Curve (AUC) of a given predictive model for different groups of students. They used their metric to assess gender-based (male vs. female) differences in classification performance of MOOC dropout models, showing that ABROCA can capture unfair classification performance related to the gender imbalance in the data. ABROCA was also used in other educational studies, to evaluate the fairness across different sociodemographic groups of classifiers meant to predict college graduation [18], and categorize students' educational forum posts [30]. The other fairness studies in education have used more common statistical metrics in AI, such as group fairness, equalized odds, equal opportunity, true positive rate and false positive rate between groups, to predict course completion [26], at-risk students [17], and college grades and success [19, 36, 25]. Similarly to ABROCA, these metrics are predictive performance-oriented. In this paper, we contribute to this line of fairness work in education by investigating the possibility and value of a fairness metric that accounts for the behaviors of the classifiers.

3. AN ALGORITHMIC FAIRNESS ANALYSIS APPROACH

3.1 Definition of the MADD metric

We introduce a novel metric, the Model Absolute Density Distance (MADD), which is based on measuring algorithmic fairness via models' behavior differences between two groups, instead of via models' predictive performance. It is worth noting that this focus on the models' behaviors enables us to not only quantify algorithmic fairness, but also gain a deeper understanding of how the models discriminate via graphical representations of the MADD (see next subsection 3.2).

We present the MADD under the scope of this study where we consider binary sensitive features and binary classifiers that output probability estimates (or confidence scores) associated to their predictions.

Assume a model $\mathcal{M}$, trained on a dataset $\{X, S, Y\}_{i=1}^{n}$ where S are the binary sensitive features, X all the other features characterizing the students, $Y \in \{0, 1\}$ the binary target variable, and n the number of samples. $\{X, S\}_{i=1}^{n}$ represents all the features of the prediction task. More precisely, $S = (s_i^a)_{i=1}^{n}$ where a is the index of the considered sensitive feature and $s_i^a \in \{0, 1\}$. Indeed, if a student (x_i, s_i^a) belongs to any group named G_0 of the sensitive feature a, then $s_i^a = 0$, and idem $s_i^a = 1$ if (x_i, s_i^a) belongs to the other group named G_1 of the same sensitive feature a. Note that a sample (x_i, s_i^a) describes a unique student in a group, with the groups G_0 and G_1 being mutually exclusive (i.e. a student can only belongs to one of these two groups). Also, none of these groups is considered as a baseline or privileged group here.

$\mathcal{M}$ aims at minimizing some loss function $\mathcal{L}(Y, \hat{Y})$ with its predictions $\hat{Y}$ to estimate or predict Y. $\mathcal{M}$ should assign to each $\hat{Y}_i$ a predicted probability (or a confidence score) that a given sample (x_i, s_i^a) will be predicted as $\hat{Y}_i = 1$. This probability or score is noted $\hat{p}(x_i, s_i^a) = \mathrm{P}(Y = 1 | X_i = x_i, S = s_i^a)$. We introduce a parameter e that is the probability sampling step of $\hat{p}$ values between 0 and 1. In other words, $\hat{p}$ values are rounded to the nearest e (e.g. $\hat{p}(x_i, s_i^a) = 0.09$ if $e = 0.01$ and the same $\hat{p}(x_i, s_i^a) = 0.1$ if $e = 0.1$ for instance). $\mathcal{M}$ predicts $\hat{Y}_i = 1$ if and only if $\hat{p}(x_i, s_i^a) \geq t$ where t is a probability threshold, and $\hat{Y}_i = 0$ otherwise.

We define two unidimensional vectors $D_{G_0}^a$ and $D_{G_1}^a$ as what we call in short the *density vectors* of the respective groups G_0 and G_1 of the sensitive feature a. They actually contain all the density values associated to each $\hat{p}(x_i, s_i^a)$ value (rounded to the nearest e) of group G_0 or group G_1. In particular, $D_{G_0}^a = (d_{G_0,k}^a)_{k=0}^{m}$ where each $d_{G_0,k}^a$ is the density of $\hat{p}(x_i, s_i^a) = k \times e$ value, that is to say the frequency that the model $\mathcal{M}$ gives $\hat{p}(x_i, s_i^a) = k \times e$ divided by the sum of frequency of all $\hat{p}$ values. m is equal to the total number of distinct $\hat{p}$ values and is related to e by the following: $m = 1/e + 1$. The advantage of the introduction of e could be seen here: having discretized the $\hat{p}$ values enables us to have the two density vectors $D_{G_0}^a$ and $D_{G_1}^a$ of the same length so that they are comparable on the probability space and independent from the model $\mathcal{M}$'s behaviors.

We now define the MADD as follows:

$$\mathrm{MADD}(D_{G_0}^a, D_{G_1}^a) = \sum_{k=0}^{m} |d_{G_0,k}^a - d_{G_1,k}^a| \tag{1}$$

The MADD satisfies the necessary properties of a metric: reflexivity, non-negativity, commutativity, and triangle inequality [9] (see the proofs in Appendix A). Moreover:

$$\forall a, \quad 0 \leq \mathrm{MADD}(D_{G_0}^a, D_{G_1}^a) \leq 2 \tag{2}$$

The closer the MADD is to 0, the fairer the outcome of the model is regarding the two groups. Indeed, if the model produces the same probability outcomes for both groups, then $D_{G_0}^a = D_{G_1}^a$ and $\mathrm{MADD}(D_{G_0}^a, D_{G_0}^a) = 0$. Conversely, in the most unfair case, where the model produces totally distinct probability outcomes for both groups, the MADD is equal to 2. An example of such a situation is when $\exists k_{G_0}, d_{G_0,k_{G_0}}^a = 1$ and $\forall k \in [0, m], k \neq k_{G_0}, d_{G_0,k}^a = 0$, and $\exists k_{G_1} \neq k_{G_0}, d_{G_1,k_{G_1}}^a = 1$ and $\forall k \in [0, m], k \neq k_{G_1}, d_{G_1,k}^a = 0$. In that case, Equation 1 becomes:

$$\mathrm{MADD}(D_{G_0}^a, D_{G_1}^a) = |d_{G_0,k_{G_0}}^a| + |d_{G_1,k_{G_1}}^a| = (1 + 1) = 2 \tag{3}$$

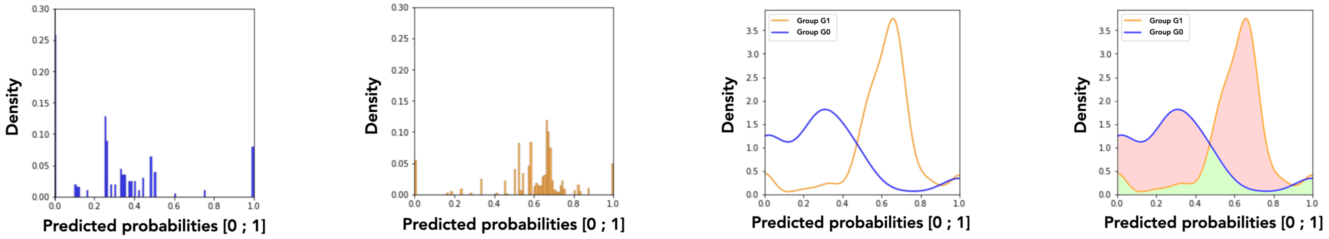
3.2 Visualization-based analysis of models' discriminatory behaviors

We introduce a visualization-based analysis of the models' discriminatory behaviors that complements our fairness analysis approach. This analysis is based on graphical interpretations of the MADD. Let us plot in Figures 1a and 1b the density histograms associated with each density vector $D_{G_0}^a$ and $D_{G_1}^a$. These histograms represent the distributions of the $\hat{p}$ values for the group G_0 and the group G_1 of a sensitive feature a. The number and consequently the width of the intervals depend on the probability sampling step e.

However, these histograms are not easily interpretable because of the numerous variations of the discrete values. We solve this issue by applying a smoothing by kernel density estimation (KDE) with Gaussian kernels, as shown in Figure 1c. The smoothing parameter, also called bandwidth parameter, is determined by the Scott's rule, an automatic bandwidth selection method[2]. This smoothing transforms the discrete probability distribution (whose density values cannot exceed 1 in the y-axis as they are related to discrete random variables) into a continuous approximation of the associated probability density function (PDF), which can in turn take values greater than 1.

Therefore, a visual approximation of the MADD corresponds to the red area in Figure 1d. Indeed, as the MADD uses the absolute density distances point-by-point between the two density vectors, the metric can be visually approximated by the area in-between the two curves, considering that the graph shows continuous density instead of the true discrete values used in the MADD calculation. Conversely, the green area, which is the intersection of the smoothed representations of the two density vectors, illustrates the area where the model $\mathcal{M}$ produces the same predicted probabilities for both groups up to a certain approximated density. We call this area the *fair zone*.

[2] See documentation of `scipy.stats.gaussian_kde`.



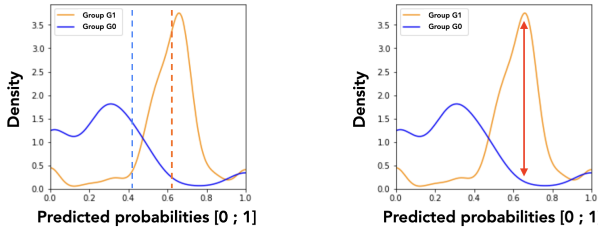
(a)

(b)

(c)

(d)

Figure 1: Visual representation of the MADD. Histograms of predicted probabilities for group G_0 (a) and group G_1 (b). Smoothing of these histograms (c). Approximation of the MADD in the red zone (d) vs. the *fair zone* in green.



(a) Unequal treatment

(b) Stereotypical judgement

Figure 2: Two models' discriminatory behaviors. The dotted lines are the respective means of the two density vectors.

Thanks to this graphical representation approximating the MADD, we are able to distinguish two model's discriminatory behaviors: unequal treatment (Figure 2a), and stereotypical judgement (Figure 2b). Unequal treatment behavior can be summarized as follows: "how much the model favor or penalize individuals based on them belonging to each group?" As displayed in Figure 2a, we can identify which group get lower or higher predicted probabilities on average, allowing us to understand which group the model tends to favor (the highest mean, here the group G_1) or to penalize (the lowest mean, here the group G_0). It is worth noticing that the means are not perfectly aligned with the peaks of the distributions because they are calculated from the density vectors, without the smoothing. The second discriminatory behavior, stereotypical judgement, can be summarized as follows: "how much the model makes repetitive and invariant "judgement" about the individuals based on them belonging to a group?" For instance in Figure 2b, the model clearly tends to give to many persons in the group G_1 the same predicted probabilities. These analyses cannot be performed with existing predictive performance-oriented fairness metrics, as the model could have the same accuracy for both groups regardless of its underlying effective predictions, either in terms of distributions or in terms of density differences.

4. EXPERIMENTAL SETTING

We apply our approach on the common task of predicting student success to a course, and we present in this section (1) the data, (2) the models, and (3) the setting parameters we used in our experiments. This case study is designed to further investigate our proposed approach, and to show how one can use it.

4.1 Data

4.1.1 Dataset presentation

We used real-world anonymized data from the Open University Learning Analytics Dataset (OULAD) [22]. The Open University is a distance learning university from the United Kingdom, offering higher education courses which can be taken as standalone courses or as part of a university program with no previous qualifications required. The dataset contains both student demographic data and interaction data with the university's virtual learning environment (VLE). The students were enrolled in at least one of the three courses in Social Sciences or one of the four Science, Technology, Engineering and Mathematics (STEM) courses between 2013 and 2014. The dataset contains 32,593 samples including 28,785 unique students.

The choice of this dataset was motivated by several reasons. First, the OULAD is one of the most comprehensive and benchmark datasets in the learning analytics domain to assess the performance of students in a VLE [1]. In addition, it is an open dataset that answers the call to the community for the development of new approaches on open datasets [15]. Then, it also answers another call from [15] for replication in multiple contexts such as several courses with diverse populations, as provided in the OULAD. Moreover, as it is commonly the case with distance learning universities, the students have a large variety of profiles [8] (including on average more women than men and a wide age range [2]), and these information are available in the dataset, making it particularly relevant for studying the impact of demographic features in terms of fairness. Finally, the data was collected in compliance with The Open University requirements regarding ethics and privacy, including consent and anonymization.

4.1.2 Data preprocessing

We used the features presented in Table 1. The `sum_click` feature was the only one that was not immediately available in the original dataset and was computed from inner joints and aggregation on the original data. Also, we removed samples where the value of the `poverty` feature was missing (4% of the data samples) and when the students withdrew from the courses (24% of the data samples). This left us with 19,964 samples of distinct students, whose values were scaled between 0 and 1 for every feature via normalization. We indeed did not apply standardization to keep the original data distributions and analyze the models' behaviors accordingly. The target variable (course outcome)

Table 1: Features used from the OULAD dataset [22].

Name	Feature type	Description
`gender`	binary	the students' gender
`age`	ordinal	the interval of the students' age
`disability`	binary	indicates whether the students have declared a disability
`highest_education`	ordinal	the highest student education level on entry to the course
`poverty`[3]	ordinal	specifies the Index of Multiple Deprivation [22] band of the place where the students lived during the course
`num_of_prev_attempts`	numerical	the number of times the students have attempted the course
`studied_credits`	numerical	the total number of credits for the course the students are currently studying
`sum_click`	numerical	the total number of times the students interacted with the material of the course

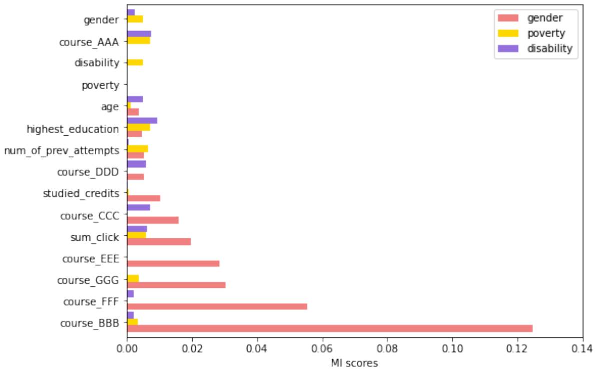


Figure 3: Mutual information (MI) scores.

was coded as "Pass" or "Fail" (1 or 0 respectively). Plus, students who got a "Distinction" outcome were also coded as 1 ("Pass"), as we target binary classification for this case study.

In our study, we considered three sensitive features: `gender`, `poverty`, and `disability`. Although other sensitive features could have been relevant, our main focus here is in investigating our proposed method itself. Therefore, one may choose different sensitive features according to their purpose (for instance including age as well, as in [34]), and one would be able to conduct the same fairness analysis process. Due to our method dealing with binary features as sensitive features, we transformed `poverty` into a binary feature by setting a 50% threshold of deprivation index [22], coding as 0 those below the 50% threshold (i.e. less deprived) and as 1 those above (i.e. more deprived).

We did not apply any data balancing techniques nor unfairness mitigation preprocessing, still to keep the original data distributions. However, our approach does not prevent the use of such preprocessing.

4.1.3 *Data analyses and course selection*

We explored the correlations and imbalances of the sensitive features across the different courses in the dataset to identify those which were relevant for analysing algorithmic fairness. We thus computed the mutual information (MI) between all the features and the three sensitive ones, whose respective results are distinguished by a different color as shown in Figure 3. Mutual information is particularly relevant for non-linear relationships between features. Figure 3 shows that the course "BBB" followed by the course "FFF" are the most correlated with the `gender` feature, with the overall highest MI scores. Therefore, the Social Sciences course coded as "BBB" and the STEM course coded as "FFF", two different student populations, were good candidates for examining the impact of gender bias on the predictive models fairness. In addition, both courses presented very high imbalances in terms of `disability` (respectively 91.2-8.8% and 91.7-8.3% for 0-1 groups in courses "BBB" and "FFF") and `gender` (respectively 88.4-11.6% and 17.8-82.2% for 0-1 groups in courses "BBB" and "FFF"), and still some imbalance for `poverty` (respectively 42.3-57.7% and 46.9-53.1% for 0-1 groups in courses "BBB" and "FFF"). Based on these preliminary unfairness expectations derived from the skews in the data, it is interesting to analyze whether and how the models will suffer from these biases in both courses. These two courses are thus excellent testbeds for testing our approach.

4.2 Classification Models

To show that our fairness analysis approach can handle several types of classification models, we chose models either based on regression, distance, trees, or probabilities. More precisely, we chose a logistic regression classifier (LR), a k-nearest neighbors classifier (KN), a decision tree classifier (DT), and a naive bayes classifier (NB).

We chose these particular models for the following reasons. Firstly, they are widely used in education, and specially with the OULAD [21, 1]. Models based on vectors (e.g. support vector machines), also commonly used, were not selected as they do not outcome probability estimates (or confidence scores) on which to run our fairness analysis. Secondly, while our approach can be generalized to other models with probability estimates (or confidence scores) such as random forest or neural networks, we favored white boxes and explainability over finding the best modeling with fine-tuning. Thirdly, predicting students' success with the data in the OULAD is a rather low abstraction task due to the small amount of features and variance in the data, for which using complex predictive models would not lead to better performance and could even overfit the data. Finally, the selected models are easy to implement for most use cases, which makes them universally good candidates for predictive modeling in general.

[3] Named as `imd_band` in the original data.

To fit the models, we split the data into a train and a test set using a 70-30% split ratio in a stratified way, meaning that we kept the same proportion of students who passed and failed in both the train and the test sets. The resulting accuracies of the models were above the baseline (70%) and up to 93%, except for the NB (62%) which instead presented interesting behaviors with the MADD analyses and was worth keeping it (see Section 5). It has to be noted that, contrary to most ML studies, achieving the best predictive performance was not our focus here, since the purpose of our experiments is rather to analyze the fairness of diverse models with the MADD metric. Then, we used the models' outcomes on the test set to compute the MADD metric and generate the visualizations.

4.3 Fairness Parameters

For our study, we set e to 0.01 (i.e. $m = 101$), and t, the probability classification threshold, to 0.5. For e, 0.01 corresponds to a variation of the probability of success or failure of 1%, which we deem a sufficient level of probability sampling precision, considering on the one hand that probability variations below 1% are not significant enough in the problem, and on the other hand that higher values of e (up to 0.1) did not alter the MADD results. Regarding t, the success prediction is generally defined by having an average score above 50% and thus we chose t with respect to the problem rather than optimizing it for model performance. The odds of positive or negative predictions are thus balanced and the threshold is the same for each individual.

5. RESULTS

In the following, we show in subsection 5.1 how the MADD and its visualization-based analysis can help unveil unexpected results based on (1) the respective importance of each sensitive feature in algorithmic unfairness, (2) the models intrinsic unfairness, and (3) the nature of the unfairness associated with the predictions made by the model. Then, we show in subsection 5.2 how our results differ from and complement what can be provided by ABROCA, a state-of-the-art predictive performance-oriented fairness metric. Both subsections 5.1 and 5.2 are concluded by a summary of the obtained results.

5.1 Fairness Analysis with MADD

In the parts 5.1.1 and 5.1.2, we examine via Tables 2 and 3 the MADD results reported for the two courses. We highlight in bold the best MADD per column, and with an asterisk (*) the best MADD per row. In this way, the MADD of the fairest model for each sensitive feature is in bold, whereas the MADD of the fairest sensitive feature for each classifier is marked with a *. As examples, in Table 2 the DT is the fairest model regarding the `poverty` feature (bold), and in Table 3 the `disability` feature is the fairest for the KN (*). For the part 5.1.3, we base our visual analyses and identification of discriminatory behaviors explained in subsection 3.2 on Figures 4 and 5.

5.1.1 Sensitive features analysis

Course "BBB" (Social Sciences). Table 2 reveals that three models out of four (LR, KN, and DT) are the fairest for the `disability` sensitive feature. Therefore, two interesting observations can be made. First, it is contrary to what we would expect since `disability` was the most imbalanced (91.2-8.8% for 0-1 groups) sensitive feature in the training data (see Section 4.1.3). Second, the `gender` feature was particularly expected to be highly sensitive due to its high correlation with the target in this course and its imbalance, but it actually has the best MADD on average (1.02).

Course "FFF" (STEM). Similarly to the above results for the course "BBB", we can notice that the data skews are not necessarily reflected in the MADDs. In the training data, the `disability` sensitive feature was highly imbalanced, and the `poverty` feature was quite balanced. Nonetheless, for half of the models (see Table 3), both `disability` and `poverty` are the two sensitive features with regard to which the models are the fairest. On the other hand, in line with the gender skew and correlation shown in Section 4.1.3, `gender` has the worst MADD results in average, more than `disability`, although the difference is not substantial.

5.1.2 Model fairness analysis

Course "BBB" (Social Sciences). Now focusing on the fairness of the models, DT appears in Table 2 to be the fairest, with an average MADD of 0.73 across all the sensitive features. DT is indeed the fairest for `disability` and `poverty` and the second best for `gender`. On the contrary, LR is the least fair, with the highest results for each senstive feature and an average of 1.71, with a maximum value of 2.

Course "FFF" (STEM). NB and DT obtain the best MADD averages across all three sensitive features (0.64 and 0.65 respectively). Therefore, there is no clear winner for this course as they behave differently according to different sensitive features: NB has better results for `gender` and `poverty` but a higher MADD for `disability`, whereas DT is more balanced across the three sensitive features. However, we remind that NB performed below the accuracy baseline and thus DT would overall be a better candidate.

Table 2: MADD results for the course "BBB".

	Model	Sensitive features			
		`gender`	`poverty`	`disability`	Average
MADD	LR	1.72	1.85	1.57*	1.71
	KN	1.13	1.12	0.93*	1.06
	DT	0.69	**0.85**	**0.65***	0.73
	NB	**0.52***	0.9	1.37	0.93
	Average	1.02	1.18	1.13	

Table 3: MADD results for the course "FFF".

	Model	Sensitive features			
		`gender`	`poverty`	`disability`	Average
MADD	LR	1.18	1.06*	1.12	1.12
	KN	1.06	0.93	0.78*	0.92
	DT	0.76	0.65	**0.55***	0.65
	NB	**0.56**	**0.47***	0.90	0.64
	Average	0.89	0.78	0.84	

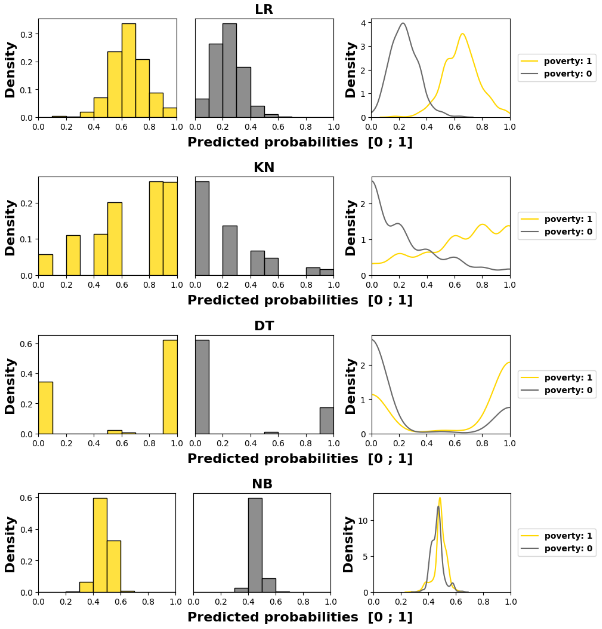


Figure 4: Models' behaviors in course "BBB". Note that for these graphs e was set to 0.1 for better visualization of the bars, but e was actually equal to 0.01 for computation, as said in subsection 4.3.

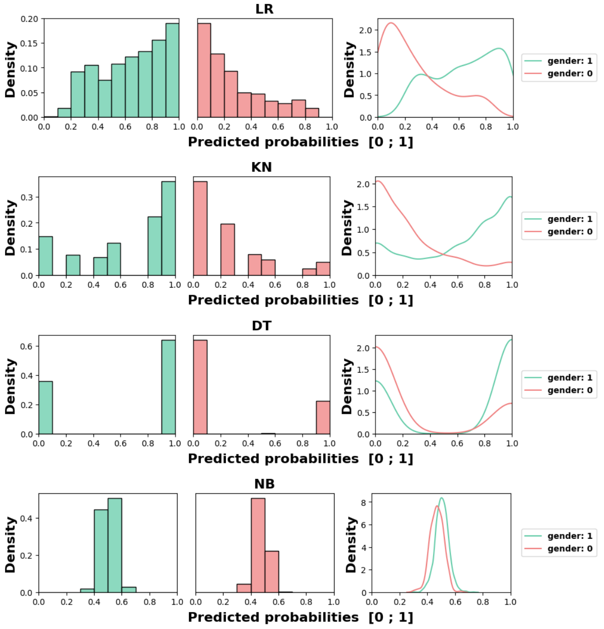


Figure 5: Models' behaviors in course "FFF". Note that for these graphs e was set to 0.1 for better visualization of the bars, but e was actually equal to 0.01 for computation, as said in subsection 4.3.

5.1.3 Visualization-based analysis

Course "BBB" (Social Sciences). We examine in Figure 4 the models' behaviors regarding the most sensitive feature, namely `poverty` in this course, as it has the worst MADD on average (1.18). We see in the subfigures, through an offset of the distribution mean to the left for the group 0, that three models out of four (LR, KN, and DT) have learned unequal treatment against those in better financial conditions (group 0). Among them, KN and DT present the highest stereotypical results reduced to only few probability values, which illustrates well their inner workings. Conversely, NB produces the least discriminating results with the closest means for the two groups. Its behavior is all the more interesting since it shows that having poor predictive performance is not necessarily interfering with behaving fairly regarding two groups of the most sensitive feature. It is precisely because it does not discriminate against any features, whether they were sensitive or not, that it has poor accuracy.

Course "FFF" (STEM). Likewise, we examine in Figure 5 the models' behaviors for the most sensitive feature in this course, `gender`, with the worst MADD average of 0.89. All models but NB exhibit unequal treatment against group 0, here the women. Similarly to the previous results, we can again note the highly stereotypical behaviors of KN and DT, and the relative fairness of the NB model.

5.1.4 Implications for the MADD

Following the double reading of the tables, feature-wise or model-wise, as well as our visual analyses, we can make two important observations regarding the insights provided by the MADD.

Firstly, there is no direct relationships between biases in the data (imbalanced representations, high correlations) and the discriminatory behaviors learned by the models. We even observe opposite conclusions (specially for the course "BBB" in part 5.1.1).

Secondly, trained on the same data, the models exhibit very different discriminatory behaviors (see parts 5.1.2 and 5.1.3), both regarding different sensitive features, and different severity and nature of their algorithmic unfairness. This was also shown by our visual analysis, which allowed finer-grained interpretations of the discriminatory behaviors.

5.2 Comparison with ABROCA

We now aim to compare the MADD with the ABROCA predictive performance-oriented fairness metric [14]. The ABROCA results (computed with the source code from [12]) are displayed in Tables 4 and 5, and an illustrated example for the course "BBB" is given through the Figures 6 and 7.

5.2.1 Sensitive features analysis

Course "BBB" (Social Sciences). Let us first focus on the `poverty` feature which has the worst MADD average (1.18 from Table 2). In particular, in part 5.1.1, `poverty` was the feature with which LR obtained the worst MADD (1.85 from Table 2), which was also the worst MADD overall. Indeed, in Figure 6 it can be seen that LR has the smallest intersection area compared to the other models. However, in Figure 7 and Table 4 we see that LR has one of the best

ABROCA (0.03) with minimal area between the curves of the respective groups. We found similar opposite results between MADD and ABROCA for `gender`. Thus, `poverty` and `gender` could be seen as unfair sensitive features for a model on the one hand (MADD) and as fair ones on the other hand (ABROCA). Moreover, DT too has one of the best ABROCA (0.03), while it provided the best MADD value (0.85 from Table 2) regarding this feature. Therefore, two models with the same ABROCA lead to opposite discriminatory behaviors according to the MADD. In the end, ABROCA and MADD do not highlight the same fairness results, and can even lead them to show opposite results.

Course "FFF" (STEM). Now examining Table 5 for the course "FFF", ABROCA does not capture substantial differences at the sensitive feature level (column-wise), with an ABROCA average of 0.4 for all three features. Thus, the MADD results can capture additional differences among the models' behaviors that are not reflected in the ABROCA results. In addition, we again found that `disability`, the most imbalanced sensitive feature, is actually the feature for which DT is the fairest, regardless of whether we consider the MADD in part 5.1.1 (Table 3) or the ABROCA (Table 5). Therefore, ABROCA does not reflect the imbalance bias in the data either, in contradiction with the findings from [14].

5.2.2 Model fairness analysis

Course "BBB" (Social Sciences). In Table 4, LR and NB appear to be the fairest models across all the sensitive features (best common ABROCA average of 0.3). However, with the MADD, NB indeed exhibited overall quite balanced low values, but LR was always the least fair on average (Tables 2 and 3). Thus, at the model level this time, the trends in the MADD results are only partially reflected in the ABROCA results.

Course "FFF" (STEM). Table 5 shows that ABROCA results do not exhibit substantial variability to distinguish differences in fairness between the models (row-wise this time) in our experiment. In addition to similar ABROCA averages, all models have very close ABROCA results specially regarding the `gender` and `poverty` features. Therefore, the MADD allows to find complementary discriminatory results as compared to using only ABROCA.

5.2.3 Summary of the comparison

Two main takeaways could be reported from our comparison between the ABROCA and MADD metrics.

Firstly, fair predictive performance (i.e. similar numbers of errors across groups, here captured by low ABROCA values) does not guarantee fair models' behaviors (i.e. low severity of discrimination across groups, here captured by low MADD values). This demonstrates what we advocated in the introduction (Section 1) regarding investigating the models' behaviors to gain a comprehensive understanding of the models fairness. In particular, two models with the same ABROCA could suffer from substantial, and even opposite algorithmic discriminatory behaviors, which can be uncovered by the MADD (see parts 5.2.1 and 5.2.2). Using the MADD together with a predictive performance-oriented metric such as ABROCA can thus allow more informed selection of fair models in education, and here in our experiments, they provide strong evidence that DT is the fairest model on both courses.

Table 4: ABROCA results for course "BBB".

	Model	Sensitive features			
		`gender`	`poverty`	`disability`	Average
ABROCA	LR	0.02	0.03	0.03	0.03
	KN	0.08	0.06	0.06	0.07
	DT	0.06	0.03	0.05	0.05
	NB	0.04	0.02	0.04	0.03
	Average	0.05	0.04	0.05	

Table 5: ABROCA results for course "FFF".

	Model	Sensitive features			
		`gender`	`poverty`	`disability`	Average
ABROCA	LR	0.04	0.03	0.03	0.03
	KN	0.04	0.05	0.04	0.04
	DT	0.05	0.04	0.01	0.03
	NB	0.03	0.03	0.07	0.04
	Average	0.04	0.04	0.04	

Secondly, in line with our previous findings that biases in the data may not be related with models' discriminatory behaviors (see part 5.1.4), we also observed that the biases in the data are independent from predictive performance biases too. For instance, the highest imbalanced sensitive feature could actually lead to both the best ABROCA and the best MADD. Although this observation is aligned with the findings in [11, 17], it is worth noting that it goes against what the authors of ABROCA had observed [14] (see part 5.2.1).

6. DISCUSSION

In this section, we discuss (1) the overall implications of the results of our fairness study, (2) the limitations and the strengths of the proposed approach, (3) some potential experimental improvements, and (4) some guidelines to use our fairness analysis approach with the MADD.

6.1 Fairness results

Our results lead to three main conclusions, as follows. Firstly, we found no direct relationships between data bias and predictive performance bias nor discriminatory behaviors bias. It confirms previous findings that unfair biases are not only captured in the data, but are inherent to the model too [27, 28]. It further suggests that exclusively mitigating unfairness in the data might not be sufficient, and that mitigating unfairness at the model level is key too. Secondly, even trained on the same data, each model exhibits its own discriminatory behavior (likely linked to its inner working) and according to different sensitive features. It raises interesting questions on how different models could be combined in order to balance discriminatory behaviors with regards to multiple sensitive features at the same time. Thirdly, fair predictive performance does not guarantee by itself fair models' behaviors and thus fair outcomes. Additional introspection of the model is therefore needed, and our approach

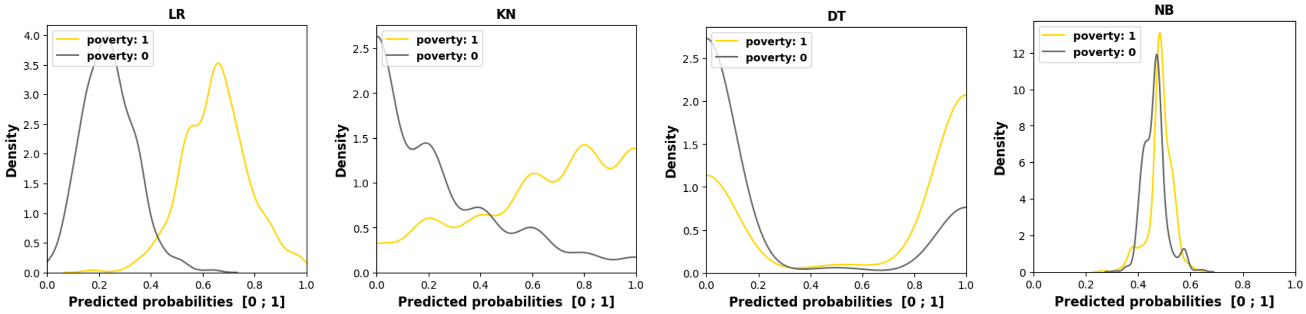


Figure 6: MADD visualizations for the `poverty` sensitive feature across all the models for course "BBB".

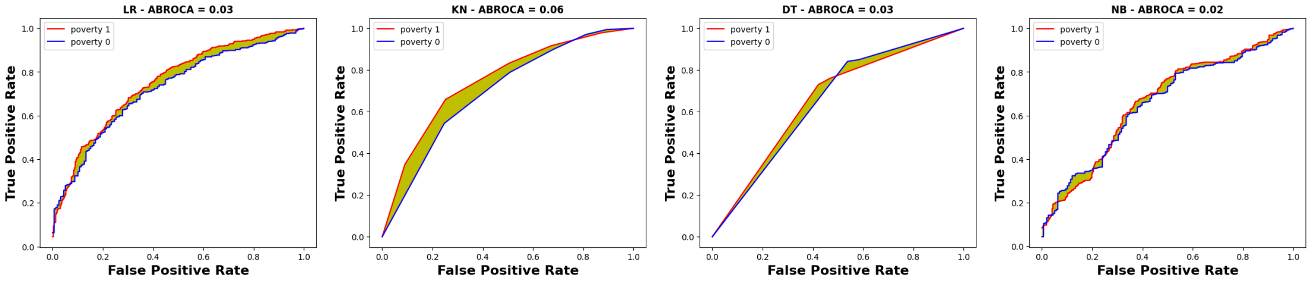


Figure 7: ABROCA slide plots for the `poverty` sensitive feature across all the models for course "BBB".

appears as a possible solution.

6.2 Limitations and strengths

Although our approach was initially designed for analyzing algorithmic fairness at an individual sensitive feature level, our prospective work includes a generalization of the MADD metric to capture the influence of multiple sensitive features simultaneously. Moreover, the current MADD is particularly suitable for binary sensitive features and binary classifiers, and future work should also focus on extending it to multi-class features and classifiers. As an example, an extension for categorical sensitive features would enable us to have a finer-grained analysis of discrimination across more relevant subgroups. Despite these current limitations, the strengths of our approach stand in (1) its ability to be used with any tabular data, the most prevalent data representation [24] from any domain, and without needing any unfairness mitigation preprocessing; (2) being able to have a richer understanding of models' discriminatory behaviors and their quantification with an easy-to-implement fairness metric that is independent from predictive performance; and (3) since the MADD is bounded, being comparable between different datasets to measure the discriminatory influence of a particular sensitive feature in different contexts and populations.

6.3 Experimental improvements

In our experiments we purposely focused on the MADD results to highlight its contribution and interest to fairness analysis, however for real-case applications one should obviously pay attention to both predictive performance and fairness performance in order to thoroughly select satisfying models. As an example, the NB model used in our experiments could be seen as fair regarding its MADD results, but it had in fact poor accuracy particularly because it was unable to predict well the success or the failure of students regarding any features, which makes this model not usable for real-case purposes but nonetheless interesting for our exploratory analysis. We thus recommend using the MADD on models that show satisfying predictive performance, to gain a finer-grained understanding on how they behave and regarding who and to refine models selection and their usage. Moreover, one should also consider testing variations of the probability sampling parameter e in their application and context. Although the impact of its variation in the range from 0.01 to 0.1 was low in our experiments, it might not be always the case. Determining the optimal value for this parameter is also a key part of our prospective work. Finally, we have demonstrated the validity and value of our approach on two courses of the OULAD dataset. Nonetheless, in a broader context of investigating model unfairness, this work should be replicated with other educational datasets providing more students data and more diverse sensitive features [3].

6.4 Guidelines

In order to facilitate replication studies and the use of our approach (in addition to the availability of the data and our source code), we provide in the following a 7-step guide to help readers compute the MADD and plot the models' behaviors as in Figures 4 and 5.

1. Choose binary classification models that can output probability estimates or confidence scores.
2. Transform, when needed, every sensitive feature into binary one.
3. Train the models, and in the testing phase separate

their predicted probabilities or confidence scores according to the groups of each sensitive feature.

4. Compute the MADD for each sensitive feature, and compare the results between features and models.
5. Plot histograms of the predicted probability distributions of each group of the sensitive features, and their smoothed estimations (e.g. with KDE).
6. Visually identify discriminatory behaviors among unequal treatment (i.e. distance between the two distribution means) and stereotypical judgement (i.e. differences of local amplitudes).
7. Depending on the fairness analysis goals:
    - Identify which models are the fairest overall or according to which sensitive features, using a row-wise reading of the results table.
    - Identify which features are the most sensitive overall or according to which models, using a column-wise reading of the results table.
    - Using the plots, identify which groups (i.e. which distributions) are the most discriminated against by the models (relatively to each sensitive feature).

7. CONCLUSION

In this paper, we developed an algorithmic fairness analysis approach based on a novel metric, the *Model Absolute Density Distance* (MADD). It measures models' discriminatory behaviors between groups, independently from their predictive performance. Our results on the OULAD dataset and comparison with ABROCA show that (1) fair predictive performance does not guarantee fair models' behaviors and thus fair outcomes, (2) there is no direct relationships between data bias and predictive performance bias nor discriminatory behaviors' bias, and (3) trained on the same data, models exhibit different discriminatory behaviors and according to different sensitive features.

This approach, for which we provide a set of guidelines in subsection 6.4 and our source code and data in open access at `https://github.com/melinaverger/MADD`, can be used to help identify fair models, exhibit sensitive features, and determine students who were the most discriminated against and how (unequal treatment or stereotypical judgement) in an education context. Being bounded, an advantage of this metric is that it can be used across different contexts and data to discover the features that more generally cause algorithmic discrimination.

Future work will involve the generalization of the MADD metric to multiple sensitive features, its extension to multi-class sensitive features and classifiers, determining the optimal probability sampling parameter, and we will investigate how to use the MADD as an objective function to optimize models accordingly (in addition to predictive performance objectives).

8. ACKNOWLEDGMENTS

The authors want to thank Chunyang Fan (Sorbonne Université) for his valuable comments on this work.

9. REFERENCES

[1] H. A. Alhakbani and F. M. Alnassar. Open Learning Analytics: A Systematic Review of Benchmark Studies using Open University Learning Analytics Dataset (OULAD). In *2022 7th International Conference on Machine Learning Technologies (ICMLT)*, ICMLT 2022, pages 81–86, New York, NY, USA, June 2022. Association for Computing Machinery.

[2] C. B. Aslanian and D. L. Clinefelter. Online college students 2012. *The Learning House, Inc. and EducationDynamics*, 2012.

[3] R. S. Baker and A. Hawn. Algorithmic Bias in Education. *International Journal of Artificial Intelligence in Education*, 32:1052–1092, Nov. 2021.

[4] T. Bolukbasi, K.-W. Chang, J. Zou, V. Saligrama, and A. Kalai. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In *30th Conference on Neural Information Processing Systems (NIPS 2016)*, pages 4349–4357, July 2016.

[5] J. Buolamwini and T. Gebru. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, pages 77–91. PMLR, Jan. 2018. ISSN: 2640-3498.

[6] S. Burgstahler. Opening Doors or Slamming Them Shut? Online Learning Practices and Students with Disabilities. *Social Inclusion*, 3(6):69–79, Dec. 2015.

[7] A. Castelnovo, R. Crupi, G. Greco, D. Regoli, I. G. Penco, and A. C. Cosentini. A clarification of the nuances in the fairness metrics landscape. *Scientific Reports*, 12(1):1–21, 2022.

[8] J. Castles. Persistence and the adult learner: Factors affecting persistence in open university students. *Active Learning in Higher Education*, 5(2):166–179, 2004.

[9] S.-H. Cha and S. N. Srihari. On measuring the distance between histograms. *Pattern Recognition*, 35:1355–1370, 2002.

[10] J. Dastin. Amazon scraps secret AI recruiting tool that showed bias against women, Oct 2018. Reuters.

[11] O. B. Deho, C. Zhan, J. Li, J. Liu, L. Liu, and T. Duy Le. How do the existing fairness metrics and unfairness mitigation algorithms contribute to ethical learning analytics? *British Journal of Educational Technology*, 53(4):822–843, 2022.

[12] V. Disha. https://pypi.org/project/abroca (v. 0.1.3).

[13] L. Foreman-Murray, S. Krowka, and C. E. Majeika. A systematic review of the literature related to dropout for students with disabilities. *Preventing School Failure: Alternative Education for Children and Youth*, 66(3):228–237, July 2022.

[14] J. Gardner, C. Brooks, and R. Baker. Evaluating the Fairness of Predictive Student Models Through Slicing Analysis. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 225–234, Tempe AZ USA, Mar. 2019. ACM.

[15] A. Hellas, P. Ihantola, A. Petersen, V. V. Ajanovski, M. Gutica, T. Hynninen, A. Knutas, J. Leinonen, C. Messom, and S. N. Liao. Predicting academic performance: A systematic literature review. In *Proceedings Companion of the 23rd Annual ACM*

Conference on Innovation and Technology in Computer Science Education, ITiCSE 2018 Companion, page 175–199, New York, NY, USA, 2018. Association for Computing Machinery.

[16] K. Holstein and S. Doroudi. Equity and Artificial Intelligence in Education: Will "AIEd" Amplify or Alleviate Inequities in Education? *arXiv:2104.12920*, 2021.

[17] Q. Hu and H. Rangwala. Towards Fair Educational Data Mining: A Case Study on Detecting At-risk Students. *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, page 7, 2020.

[18] S. Hutt, M. Gardner, A. L. Duckworth, and S. K. D'Mello. Evaluating fairness and generalizability in models predicting on-time graduation from college applications. *International Educational Data Mining Society*, 2019.

[19] W. Jiang and Z. A. Pardos. Towards equity and algorithmic fairness in student grade prediction. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 608–617, 2021.

[20] R. F. Kizilcec and H. Lee. Algorithmic Fairness in Education. *arXiv:2007.05443 [cs]*, Apr. 2021.

[21] C. Korkmaz and A.-P. Correia. A review of research on machine learning in educational technology. *Educational Media International*, 56(3):250–267, 2019.

[22] J. Kuzilek, M. Hlosta, and Z. Zdrahal. Open university learning analytics dataset. *Sci Data 4*, 170171, 2017.

[23] J. Larson, S. Mattu, L. Kirchner, and J. Angwin. How We Analyzed the COMPAS Recidivism Algorithm. *ProPublica*, 2016.

[24] T. Le Quy, A. Roy, V. Iosifidis, W. Zhang, and E. Ntoutsi. A survey on datasets for fairness-aware machine learning. *WIREs Data Mining and Knowledge Discovery*, 12(3):e1452, 2022.

[25] H. Lee and R. F. Kizilcec. Evaluation of Fairness Trade-offs in Predicting Student Success. *arXiv:2007.00088 [cs]*, June 2020.

[26] C. Li, W. Xing, and W. Leite. Yet another predictive model? Fair predictions of students' learning outcomes in an online math learning platform. In *Proceedings of the 11th International Learning Analytics and Knowledge Conference*, pages 572–578, 2021.

[27] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A Survey on Bias and Fairness in Machine Learning. *arXiv:1908.09635 [cs]*, Jan. 2022. arXiv: 1908.09635.

[28] D. Pessach and E. Shmueli. Algorithmic Fairness. *arXiv:2001.09784 [cs, stat]*, Jan. 2020.

[29] C. Romero and S. Ventura. Educational data mining and learning analytics: An updated survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10:e1355, 2020.

[30] L. Sha, M. Rakovic, A. Whitelock-Wainwright, D. Carroll, V. M. Yew, D. Gasevic, and G. Chen. Assessing algorithmic fairness in automatic classifiers of educational forum posts. In *International conference on artificial intelligence in education*, pages 381–394. Springer, 2021.

[31] N. L. Shaheen and J. Lazar. K-12 Technology Accessibility: The Message from State Governments. *Journal of Special Education Technology*, 33(2):83–97, June 2018.

[32] F. Sovrano, S. Sapienza, M. Palmirani, and F. Vitali. A Survey on Methods and Metrics for the Assessment of Explainability under the Proposed AI Act. In *Legal Knowledge and Information Systems: JURIX 2021: The Thirty-fourth Annual Conference*, volume IOS Press, pages 235–242, Vilnius, Lithuania, 2022.

[33] J. Vasquez Verdugo, X. Gitiaux, C. Ortega, and H. Rangwala. FairEd: A Systematic Fairness Analysis Approach Applied in a Higher Educational Context. In *LAK22: 12th International Learning Analytics and Knowledge Conference*, pages 271–281, Online USA, Mar. 2022. ACM.

[34] M. Verger, F. Bouchet, S. Lallé, and V. Luengo. Caractérisation et mesure des discriminations algorithmiques dans la prédiction de la réussite à des cours en ligne. In *11ème Conférence sur les Environnements Informatiques pour l'Apprentissage Humain (EIAH 2023)*, Brest, France, June 2023.

[35] S. Verma and J. S. Rubin. Fairness definitions explained. *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*, pages 1–7, 2018.

[36] R. Yu, Q. Li, C. Fischer, S. Doroudi, and D. Xu. Towards accurate and fair prediction of college success: Evaluating different sources of student data. *International Educational Data Mining Society*, 2020.

APPENDIX

A. PROOFS FOR MADD AS A METRIC

We remind the definition of the MADD:

$$\mathrm{MADD}(D^a_{G_0}, D^a_{G_1}) = \sum_{k=0}^{m} |d^a_{G_0,k} - d^a_{G_1,k}| \tag{1}$$

The MADD satisfies the necessary properties of a metric [9]:

$$\mathrm{MADD}(D^a_{G_0}, D^a_{G_0}) = 0 \quad \text{reflexivity} \tag{4}$$

$$\mathrm{MADD}(D^a_{G_0}, D^a_{G_1}) \geq 0 \quad \text{non-negativity} \tag{5}$$

$$\begin{split}\mathrm{MADD}(D^a_{G_0}, D^a_{G_1}) = \mathrm{MADD}(D^a_{G_1}, D^a_{G_0}) \\ \text{commutativity}\end{split} \tag{6}$$

$$\begin{split}\mathrm{MADD}(D^a_{G_0}, D^a_{G_2}) \leq \mathrm{MADD}(D^a_{G_0}, D^a_{G_1}) \\ +\mathrm{MADD}(D^a_{G_1}, D^a_{G_2}) \quad \text{triangle inequality}\end{split} \tag{7}$$

Proof for reflexivity (Eq. 4)

$$\mathrm{MADD}(D^a_{G_0}, D^a_{G_0}) = \sum_{k=0}^{m} |d^a_{G_0,k} - d^a_{G_0,k}| = 0$$

Proof for non-negativity (Eq. 5)

Due to the positivity of each term in the sum thanks to the absolute value operator, the sum of these positive terms is always positive and $\mathrm{MADD}(D^a_{G_0}, D^a_{G_1}) \geq 0$.

Proof for commutativity (Eq. 6)

Let x and y be real numbers. By commutativity of the absolute value operator, $|x - y| = |y - x|$. Thus, for any k, $|d^a_{G_0,k} - d^a_{G_1,k}| = |d^a_{G_1,k} - d^a_{G_0,k}|$ and then $\mathrm{MADD}(D^a_{G_0}, D^a_{G_1}) = \mathrm{MADD}(D^a_{G_1}, D^a_{G_0})$.

Proof for triangle inequality (Eq. 7)

Let x and y be real numbers. By triangle inequality of the absolute value operator, $|x+y| \leq |x|+|y|$. Let $x = d^a_{G_0,k} - d^a_{G_1,k}$ and $y = d^a_{G_1,k} - d^a_{G_2,k}$. Then, for any k :

$$\begin{aligned}
&|x+y| \leq |x|+|y| \\
\Leftrightarrow |(d^a_{G_0,k} - d^a_{G_1,k}) + (d^a_{G_1,k} - d^a_{G_2,k})| &\leq |d^a_{G_0,k} - d^a_{G_1,k}| \\
&\quad +|d^a_{G_1,k} - d^a_{G_2,k}| \\
\Leftrightarrow |d^a_{G_0,k} - d^a_{G_2,k}| &\leq |d^a_{G_0,k} - d^a_{G_1,k}| + |d^a_{G_1,k} - d^a_{G_2,k}|
\end{aligned}$$

Then, by linearity of the sum :

$$\begin{aligned}
\sum_{k=0}^{m} |d^a_{G_0,k} - d^a_{G_2,k}| &\leq \sum_{k=0}^{m} |d^a_{G_0,k} - d^a_{G_1,k}| + \\
&\quad \sum_{k=0}^{m} |d^a_{G_1,k} - d^a_{G_2,k}| \\
\Leftrightarrow \mathrm{MADD}(D^a_{G_0}, D^a_{G_2}) &\leq \mathrm{MADD}(D^a_{G_0}, D^a_{G_1}) + \\
&\quad \mathrm{MADD}(D^a_{G_1}, D^a_{G_2})
\end{aligned}$$

Evaluating Quadratic Weighted Kappa as the Standard Performance Metric for Automated Essay Scoring

Afrizal Doewes[1,2], Nughthoh Arfawi Kurdhi[2], Akrati Saxena[1,3]
[1]Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands
[2]Universitas Sebelas Maret, Indonesia
[3]Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands
a.doewes@tue.nl, arfa@mipa.uns.ac.id, a.saxena@liacs.leidenuniv.nl

ABSTRACT

Automated Essay Scoring (AES) tools aim to improve the efficiency and consistency of essay scoring by using machine learning algorithms. In the existing research work on this topic, most researchers agree that human-automated score agreement remains the benchmark for assessing the accuracy of machine-generated scores. To measure the performance of AES models, the Quadratic Weighted Kappa (QWK) is commonly used as the evaluation metric. However, we have identified several limitations of using QWK as the sole metric for evaluating AES model performance. These limitations include its sensitivity to the rating scale, the potential for the so-called "kappa paradox" to occur, the impact of prevalence, the impact of the position of agreements in the diagonal agreement matrix, and its limitation in handling a large number of raters. Our findings suggest that relying solely on QWK as the evaluation metric for AES performance may not be sufficient. We further discuss insights into additional metrics to comprehensively evaluate the performance and accuracy of AES models.

Keywords

Quadratic Weighted Kappa, Performance Metric, Automated Essay Scoring

1. INTRODUCTION

As the use of computer software tools for evaluating student essays becomes increasingly popular, researchers have turned to Automated Essay Scoring (AES) systems as a way to expedite the process and reduce costs. These systems, which are essentially machine learning models trained on datasets containing essay answers and their corresponding human-annotated scores, are designed to eliminate concerns about rater consistency and increase the speed of evaluation. To assess the performance of an AES system, the score predicted by an automated scorer is compared to the ground truth or the score assigned by human annotators. One common metric used to measure the accuracy of a machine learning model is the percent agreement between the predicted score and the ground truth. However, this metric has been criticized for its inability to account for chance agreement, as pointed out by Jacob Cohen in 1960 [7]. In response, Cohen developed the concept of Cohen's kappa, which takes into consideration the possibility that raters may guess certain variables due to uncertainty. To further address this issue, the variation of Cohen's kappa known as weighted kappa considers the severity of disagreement between the predicted score and the ground truth. This is particularly important in applications where the consequences of misclassification may vary. Among the variations of weighted kappa, the quadratically weighted kappa is the most commonly used for summarizing interrater agreement on an ordinal scale [12]. This trend is also evident in the field of AES systems, where QWK is frequently employed as a standard evaluation metric, as noted in numerous studies [25, 21, 26, 22, 5, 20, 1, 28, 19].

We present a comprehensive examination of the utility of Quadratic Weighted Kappa (QWK) as an evaluation metric for automated essay scoring (AES) systems. To the best of our knowledge, this is the first work to specifically address the limitations of QWK in the context of AES. We acknowledge that some of the limitations we highlight in this paper may also apply to other fields. However, our paper specifically highlights the limitation of QWK in the AES context and emphasizes its implications for practical use, particularly with respect to the threshold for model acceptance, as discussed in [25]. Our work is motivated by the fact that previous research in AES has predominantly focused on maximizing QWK performance, and we aim to draw attention to the potential pitfalls of solely relying on QWK as a measure of model performance.

While kappa statistic has proven to be effective in many cases, it has been found to have some paradoxes in certain scenarios [24, 4, 18, 27]. In a study by Brenner and Kliebsch, the sensitivity of Quadratic Weighted Kappa (QWK) to ratings (based on a given rating scale) was identified as a notable characteristic of the metric [3]. This issue is of particular relevance in our work as we delve into the implications of this characteristic on the acceptance decision of an Automated Essay Scoring (AES) model. Specifically, we focus on the impact of score resolution methods in situations where two human raters are involved in the grading

A. Doewes, N. A. Kurdhi, and A. Saxena. Evaluating quadratic weighted kappa as the standard performance metric for automated essay scoring. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 103–113, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115784

process. Standard methods for combining human scores include summing or averaging the scores. However, in the ASAP (Automated Student Assessment Prize) competition dataset, another score resolution method is employed for some prompts, which involves selecting the higher of the two scores. Our findings indicate that the treatment of human scores, despite the scores remaining unchanged, can affect the performance of the quadratic weighted kappa and ultimately influence the decision-making process regarding the acceptance or rejection of an essay scoring model. To address this issue, we also experiment with different weights on the kappa statistics in an effort to mitigate the impact of the rating scale on the kappa statistics.

Furthermore, another paradox of kappa statistics is the impact of prevalence on kappa for 2x2 agreement table that has been investigated in prior literature, as demonstrated by Byrt et al. [4]. According to them, the value of kappa is affected by the relative probability of the classes, known as the Prevalence Index (PI). When the PI is high, kappa tends to decrease, potentially leading to an underestimation of the degree of agreement between raters. However, in the context of essay examinations, binary grading or scoring systems with only two levels are relatively uncommon. Instead, grading processes typically incorporate multiple levels or categories of assessment. While the prevalence of agreement matrices with a size of 2x2 has been previously studied, there is still a lack of a comprehensive formula for calculating the prevalence of matrices with a size of 3x3 or greater. In this paper, we aim to address this gap in the literature by proposing a formula for measuring the proportions of classes in raters' agreement for agreement matrices with a size greater than 2x2.

Subsequently, our study found that the relationship between prevalence and kappa, as previously outlined by Byrt et al. [4], does not consistently hold true when applied to agreement matrices larger than 2x2. Specifically, when the prevalence index (PI) is high, the value of kappa can either decrease or increase depending on the position of the number in the diagonal of the matrix, which indicates the agreement between the two raters. It highlights the need for caution when interpreting kappa values in the context of larger matrices, such as those used to assess essay scores, as these values may not accurately reflect the true level of agreement. Our study contributes to the existing literature on the relationship between prevalence and kappa by providing new insights into the limitations of using kappa as a measure of inter-rater agreement in the context of matrices larger than 2x2.

Finally, it is crucial to consider the limitations of kappa statistics in situations involving multiple raters. Previous research has consistently emphasized the importance of involving two or more raters to increase the reliability of scores, particularly in high-stakes testing programs that include writing essays as a measured task [9]. However, it is crucial to note that kappa statistics are incapable of assessing inter-rater agreement in such situations.

The structure of the paper is as follows. In Section 2, we provide an overview of the concept of Cohen's kappa and its various weighted forms, including QWK, and examine the interpretation of their values. In Section 3, we examine the quantitative performance acceptance criteria for AES models as outlined by Williamson et al. [25]. In Section 4, we describe the experimental setup, including the dataset, the training algorithms, and the textual features of essays used to create essay scoring models. In Section 5, we assess the performance of QWK as an evaluation metric in the context of AES in multiple scenarios. The experiment results are discussed in section 6, including all notable findings. Finally, the paper is concluded in the last section.

2. KAPPA AND WEIGHTED KAPPA

Cohen's kappa and Weighted kappa are widely used measures of inter-rater agreement that account for chance agreement and have been applied in various research fields. In this section, we discuss the concept, formula, and interpretations of these two measures.

2.1 Cohen's Kappa

Cohen's kappa, also known as unweighted kappa, is a widely utilized statistical measure used to evaluate the agreement between two independent raters in their assessment of a particular set of items. This measure was first introduced by Jacob Cohen in 1960 [7] and has since become a widely accepted method for assessing the reliability of rating scales and classification models.

One of the key features of Cohen's kappa is that it adjusts for chance agreement, meaning that it takes into account the possibility of two raters agreeing simply by chance rather than as a result of their independent assessments. This is particularly useful in situations where the raters may not have a high level of expertise or may be biased in their evaluations. By normalizing the agreement between the two raters at the baseline of random chance, Cohen's kappa allows for a more objective and reliable assessment of their agreement.

Overall, the use of Cohen's kappa allows for a more accurate assessment of the agreement between two independent raters and the performance of classification models. It allows for the reliable evaluation of the reliability of rating scales and the effectiveness of classification algorithms, providing valuable insights into the accuracy and reliability of the assessments being made.

Cohen's kappa is calculated by taking into account both the observed agreement between raters and the expected level of agreement that would be observed by chance alone. By comparing these two values, Cohen's kappa allows researchers to determine the degree to which the raters' evaluations are reliable and consistent rather than merely the result of random chance.

Cohen's kappa (unweighted) is formalized as follows:

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{1}$$

In order to assess the reliability of the ratings given by the two raters, we calculated the percentage of actually observed agreement, denoted as P_o, and the expected agreement, denoted as P_e. P_o was calculated by dividing the number of ratings that were assigned the same category by both raters by the total number of ratings. This allowed us to determine the percentage of ratings that the two raters agreed upon. P_e, on the other hand, was calculated based on the distribution of ratings across the categories. Specifically, it represented the probability that the two raters would agree on a rating by chance alone. This value was obtained by taking into account the frequency of each rating within the set of ratings given by both raters.

When the value of κ is 0, it signifies that the agreement between two raters is no greater than what would be expected by chance alone. This indicates a lack of consistency in the ratings provided by the two raters. On the other hand, a κ value of 1 indicates that the raters are in complete agreement, demonstrating a high level of consistency in their ratings. It is worth noting that in rare cases, the value of κ may be negative, indicating that the agreement between the two raters is actually lower than what would be expected by chance. Table 1 provides a guide for interpreting kappa values ranging from 0 to 1, as described in the work of Landis and Koch [16]. It is important to note that it is not possible to establish a single, universally accepted value for the statistic known as kappa. Instead, the appropriateness of any particular value of kappa depends on the level of accuracy demonstrated by the observers in question, as well as the number of codes being used to categorize the data.

Table 1: Interpretation of Kappa

Kappa	Interpretation
< 0	Less than chance agreement
0.01 - 0.20	Slight agreement
0.21 - 0.40	Fair agreement
0.41 - 0.60	Moderate agreement
0.61 - 0.80	Substantial agreement
0.81 - 1.00	Almost perfect agreement

As an unweighted measure, Cohen's kappa is particularly useful for evaluating the agreement between raters when there is no inherent hierarchy or relative importance among the categories being evaluated.

2.2 Weighted Kappa

The weighted kappa statistic is a measure of inter-rater reliability that takes into account the strength of the agreement between raters in addition to the presence of the agreement itself. In contrast, the unweighted kappa statistic simply counts the number of agreements without considering the magnitude of the difference between the ratings. The use of weighted kappa is particularly appropriate in situations where the categories being rated are not equally likely or important.

To compute weighted kappa, once the observed agreement (P_o) and expected agreement(P_e) have been calculated, they are multiplied by a weights matrix. The weights would be a decreasing function of the distance $|i - j|$, such that disagreements corresponding to adjacent categories would be assigned higher weights than those corresponding to categories that are further apart [23].

There are many different ways to weigh the kappa statistic, depending on the specific situation and the type of data being analyzed. Some common weighting schemes include linear weight and quadratic weight. Given n as the number of rating categories, the formula of the linear weight for an agreement table with size $n \times n$ is as follows:

$$w_{ij} = 1 - \frac{|i-j|}{n-1}, \tag{2}$$

And, for the quadratic weight is as follows:

$$w_{ij} = 1 - \Big(\frac{i-j}{n-1}\Big)^2, \tag{3}$$

with $w_{ij} \in [0, 1]$ and $w_{ii} = 1$ for $i, j \in \{1, 2, ..., n\}$.

Linear weighting schemes assign weights to the ratings or scores based on the difference between the ratings, with larger differences receiving lower weights. Quadratic weighting schemes, on the other hand, assign weights based on the square of the difference between the ratings, with even larger differences receiving even lower weights.

3. ACCEPTANCE OF AES MODEL

According to Williamson et al. in [25], there is an acceptance criterion that is used to evaluate the performance of automated scoring in relation to human scores when automated scoring is intended to be utilized in conjunction with human scoring. The measurement of agreement between human scores and automated scores has been a longstanding method for determining the effectiveness of automated scoring systems. This evaluation process involves comparing the automated scores to the human scores in order to determine if they satisfy a predefined threshold. In particular, the quadratic weighted kappa (QWK) between automated and human scoring must be at least .70 (rounded normally) in order to be considered acceptable.

It is important to note that the performance of automated scoring systems is highly dependent on the quality of human scoring. Therefore, it is crucial that the interrater agreement among human raters is reliable before utilizing automated scoring in conjunction with human scoring. This ensures that the automated scores will be accurate and reliable, which is essential for the effective use of automated scoring in a variety of settings.

4. EXPERIMENT SETTINGS

4.1 Dataset

In order to conduct our experiment, we utilized the Automated Student Assessment Prize (ASAP) dataset[1], which is hosted on the Kaggle platform. This dataset has been widely recognized as a valuable resource for evaluating the performance of automated essay scoring (AES) systems [17], and has thus become the standard for research in this field. The ASAP dataset comprises a collection of essays that have already been scored by human graders and includes eight different prompts with a range of possible scores for each.

[1] https://www.kaggle.com/c/asap-aes

4.2 Model Training

In order to assess the performance of our regression models, we employed a 5-fold cross-validation strategy, using 80% of the data for training and 20% for testing. Three different algorithms were utilized in our analysis: Gradient Boosting, Random Forest, and Ridge Regression. In this study, the essay features were obtained using the same methodology as described in [10]. Each essay was transformed into a 780-dimensional feature vector comprising two categories: 12 interpretable features and a 768-dimensional Sentence-BERT vector representation.

We trained separate models for each prompt within the dataset. To optimize the performance of each model, we utilized different hyper-parameter configurations for each individual model. In accordance with the established standard for evaluating automated essay scoring (AES) systems, we utilized the Quadratic Weighted Kappa (QWK) score as our evaluation metric [8, 25]. This measure allows us to compare the system-predicted scores with human-annotated scores, thereby providing a quantifiable indication of the level of agreement between the two.

5. QWK EVALUATION IN AES CONTEXT

In this section, we delved into various factors that can affect the value of Quadratic Weighted Kappa (QWK) and its implications for use in the context of Automated Essay Scoring (AES). These factors include the impact of the rating scale, the kappa paradox, the proportion of classes in rater agreement, changes in agreement position, and the number of raters involved.

5.1 The Effect of Rating Scale to QWK

In this section, we delve into the topic of the sensitivity of Quadratic Weighted Kappa (QWK) to rating scales. This particular characteristic of weighted kappa has been previously discussed by Brenner and Kliebsch in their seminal work [3]. We aim to further elaborate on the implications of this sensitivity in the context of evaluating the performance of an Automatic Essay Scoring (AES) model. The sensitivity of QWK to rating scales can be clearly demonstrated through the simple case presented in Table 2. By comparing the two examples within the table, we can observe that even a slight modification in the rating scale can result in notable changes in the QWK score. The first example presented in the table yields a QWK score of 0.50, which can be considered as indicating a moderate level of agreement. In contrast, the second example has a QWK score of 0.78, indicating a substantial level of agreement between the ratings. Importantly, both examples have the same number of agreements and disagreements. This illustrates the significant impact that the rating scale can have on the QWK score, highlighting the importance of carefully considering the rating scale when utilizing this measure of agreement.

Table 2: A simple example of rating scale's effect on QWK

	Prediction	QWK	Interpretation
Rater 1	[1, 2, 3]	0.50	Moderate agreement
Rater 2	[2, 1, 3]		
Rater 1	[1, 2, 4]	0.78	Substantial agreement
Rater 2	[2, 1, 4]		

We provide experimental results that show how an AES model performance changes when trained with a different score resolution from two human raters as the final score (label). The ASAP dataset score resolution table (Table 3) outlines the scoring method for each prompt. For Prompt 1 and Prompt 7, the score is determined by adding the scores from two raters together. For Prompt 2 and Prompt 8, the score is determined by combining the scores from an essay rubric. For Prompts 3-6, the score is determined by taking the higher score of the two raters.

Table 3: Prompts in ASAP dataset

ASAP Dataset	Score Resolution
Prompt 1	Sum of two raters
Prompt 2	Combination of essay rubric scores
Prompt 3	Higher of two raters
Prompt 4	Higher of two raters
Prompt 5	Higher of two raters
Prompt 6	Higher of two raters
Prompt 7	Sum of two raters
Prompt 8	Combination of essay rubric scores

The purpose of this experiment was to investigate the effect of score resolution on QWK scores using three different machine learning models: gradient boosting, random forest, and ridge regression. The QWK scores were calculated for six different prompts, labeled Prompt 1 through Prompt 7. We exclude prompt 2 and prompt 8 since they already have specific scoring methods which involve the combination of essay rubrics.

To further explore the impact of rating scale on QWK scores, we implemented three different score resolution methods for the six prompts used in our study. The first method involved summing the scores given by both raters. The second method involved selecting the higher score between the two raters. The third method involved calculating the mean of the scores given by both raters. According to a survey of state testing programs conducted by Johnson et al. [13], it was determined that an operational score is typically formed by summing or averaging the scores of raters, when such scores meet the agency's definition of agreement, which is generally predicated on the requirement that scores be at least adjacent. Additionally, the methodology of using the higher score of both raters was employed in the ASAP dataset in four prompts.

Based on the results Table 4, it appears that the sum of the QWK scores is consistently higher than the mean scores and the higher score of the two raters for all prompts and all three models. Our results demonstrated that the use of different score resolution methods had a significant impact on QWK scores. Additionally, it appears that the gradient boosting model consistently performs the best for all prompts. The random forest model performs slightly worse, while the ridge regression model performs the worst.

Table 4: The effect of score resolution on QWK using different algorithms

Dataset	Gradient Boosting			Random Forest			Ridge Regression		
	higher	mean	sum	higher	mean	sum	higher	mean	sum
Prompt 1	0.720	0.7143	0.7840	0.6986	0.6989	0.7776	0.6672	0.662	0.7395
Prompt 3	0.6825	0.6750	0.7016	0.6641	0.6631	0.691	0.6557	0.6569	0.6928
Prompt 4	0.7649	0.7742	0.8079	0.7303	0.7323	0.7740	0.7803	0.7804	0.8124
Prompt 5	0.8077	0.8108	0.8639	0.7926	0.7889	0.8526	0.7958	0.7971	0.8568
Prompt 6	0.7964	0.7931	0.8548	0.7637	0.7619	0.8239	0.7822	0.7923	0.on47
Prompt 7	0.7350	0.7685	0.7780	0.6836	0.7121	0.7254	0.7366	0.7722	0.7785

Table 5: The effect of score resolution on QWK using different weight of kappa

Dataset	QWK			LWK			Cohen's kappa		
	higher	mean	sum	higher	mean	sum	higher	mean	sum
Prompt 1	0.720	0.714	0.784	0.600	0.609	0.599	0.502	0.525	0.347
Prompt 3	0.682	0.675	0.702	0.596	0.589	0.543	0.519	0.515	0.324
Prompt 4	0.765	0.774	0.808	0.637	0.651	0.619	0.51	0.527	0.318
Prompt 5	0.808	0.811	0.864	0.686	0.694	0.680	0.559	0.574	0.353
Prompt 6	0.796	0.793	0.855	0.666	0.655	0.656	0.535	0.515	0.321
Prompt 7	0.735	0.768	0.778	0.520	0.545	0.548	0.175	0.179	0.089

In this study, we present an argument that the primary issue in this scenario is that the scores of the two human raters are basically unchanged. The difference is how the scores are treated to obtain the final score. The different results of QWK by using the higher, the mean, and the sum value of both scores results in inconsistencies in the decision-making process of the essay scoring model acceptance. These findings indicate that in order to maximize the quadratic weighted kappa value, one can always select the approach of summing the scores of both raters as it leads to a larger scale of scores.

Researchers and practitioners should be mindful of the potential impact of rating scale choices on the resulting QWK scores and take appropriate measures to mitigate this sensitivity. One strategy that can be employed is to decrease the weight assigned to the kappa formula. To evaluate the effectiveness of this strategy, we conducted a further experiment using the same dataset discussed in Table 4. The purpose of this experiment was to compare the results of the kappa values obtained from different weights, specifically quadratic weighted kappa, linear weighted kappa, and Cohen's kappa (unweighted). In order to ensure a fair comparison, we utilized the Gradient Boosting algorithm for all calculations as the previous result has shown it to perform better than Random Forest and Ridge Regression, as shown in Table 4.

The results of our experiment are presented in Table 5. We have examined the impact of the rating scale on the quadratic weighted kappa (QWK) and found that as the scale of the scores increases, the QWK value also increases. In contrast, our results for Cohen's kappa, an unweighted measure of inter-rater agreement, revealed an opposite trend. The last column of Table 5 illustrates that the kappa values for the sum of the scores are, in fact, lower than those for the higher or mean scores from human raters. This indicates that as the scale of the rating increases, the kappa values decrease.

The Linear Weighted Kappa (LWK) method has been demonstrated to yield the most balanced results when dealing with rating scales. In situations where the scores assigned by human raters remain consistent, the manner in which the scores are treated is inconsequential, as the results obtained from the higher score, the mean, and the sum of the scores are quite similar. LWK has been found to effectively mitigate the impact of rating scales in comparison to quadratically weighted and unweighted kappa.

The immediate consequence of selecting different weights for kappa is the need to define a new threshold for the acceptance rate of an automated essay scoring model. This is due to the fact that the threshold of 0.7, which is commonly utilized in such models, was specifically defined for the use of quadratic weighted kappa. In particular, different weights of the kappa coefficient reflect different emphases on different types of agreement or disagreement; therefore, it is crucial to adjust the threshold accordingly so that the evaluation aligns with the intended focus of the scoring system. Failure to properly define and adjust the threshold for acceptable performance can result in misinterpretation or overestimation of the system's performance. Thus, it is essential for stakeholders and decision-makers to clearly define the acceptable performance criteria prior to the implementation of an automated essay scoring system.

5.2 Kappa Paradox

The kappa paradox invalidates the common assumption that the value of kappa increases as the level of agreement in data increases. This paradox occurs when a classifier exhibits a high level of percent agreement but a low kappa score, which can be counterintuitive and potentially misleading.

The paradox arises due to an imbalanced agreement between two raters. For example, consider the case of binary classification, in which both raters mostly agree on only one class. In such a scenario, the percent agreement may be high, but the kappa score may be low due to the relatively high expected agreement.

To illustrate this phenomenon, consider the following example: suppose we have two predictions from rater A and rater B, represented by arrays A and B, respectively. Both arrays have a size of 1000, and the scores for each rater are as follows:

$$A = [5, 7, 7, 9, 8, 9, 9, 9, 9, 9, 9, 9, ..., 9]$$
$$B = [8, 6, 9, 6, 8, 9, 9, 9, 9, 9, 9, 9, ..., 9]$$

Here, the percent agreement between the two raters is 99.8%. However, the QWK is only 0.488, which is below the standard acceptance criteria of 0.7 proposed by Williamson et al. [25].

This result can be attributed to the fact that kappa is a chance-adjusted measure of agreement, which accounts for the expected agreement due to chance. In other words, kappa shows how much better a model performs compared to random predictions. In the example provided, the probability of agreement by chance is relatively high, leading to a low kappa score despite the high percent agreement.

To sum up, the kappa paradox highlights the importance of considering both percent agreement and kappa in evaluating the performance of a classification model. While percent agreement may be a simple and intuitive measure, it can be misleading when there is an imbalanced agreement between raters. On the other hand, kappa considers the expected agreement due to chance and provides a more nuanced view of the model's performance.

5.3 Proportion of Categories in Agreement

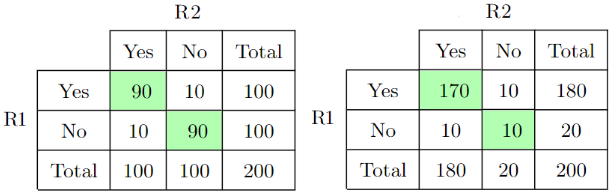
One notable limitation of QWK is that its score is heavily influenced by the proportion of agreement between raters for different classes or scores. [4] described that the value of kappa is affected by the relative probability of the classes in a 2x2 agreement table, known as the Prevalence Index (PI). Suppose there are two people who are tasked with categorizing a group of N individuals into one of two categories, such as ”Yes” or ”No”. The result can be presented in a 2-by-2 table as shown in Figure 1.

From Figure 1, the estimate of the probability of ”Yes” for the whole population would be the mean of f1/N and g1/N. Similarly, the most accurate estimate of the probability of ”No” can be obtained by finding the mean of f2/N and g2/N. The Prevalence Index (PI) is calculated by subtracting the probability of ”Yes” from the probability of ”No” and dividing the result by N. Therefore, it is estimated by (a - d)/N.

		Rater 2		
		Yes	No	Total
Rater 1	Yes	a	b	g1
	No	c	d	g2
	Total	f1	f2	N

Figure 1: Agreement table of size 2x2

The value of PI can range from - 1 to + 1, and is equal to 0 when the probabilities of ”Yes” and ”No” are equal.



		R2		
		Yes	No	Total
R1	Yes	90	10	100
	No	10	90	100
	Total	100	100	200

(a) PI = 0.0, kappa = 0.8

		R2		
		Yes	No	Total
R1	Yes	170	10	180
	No	10	10	20
	Total	180	20	200

(b) PI = 0.8, kappa = 0.44

Figure 2: Prevalence and kappa correlation on 2x2 matrix

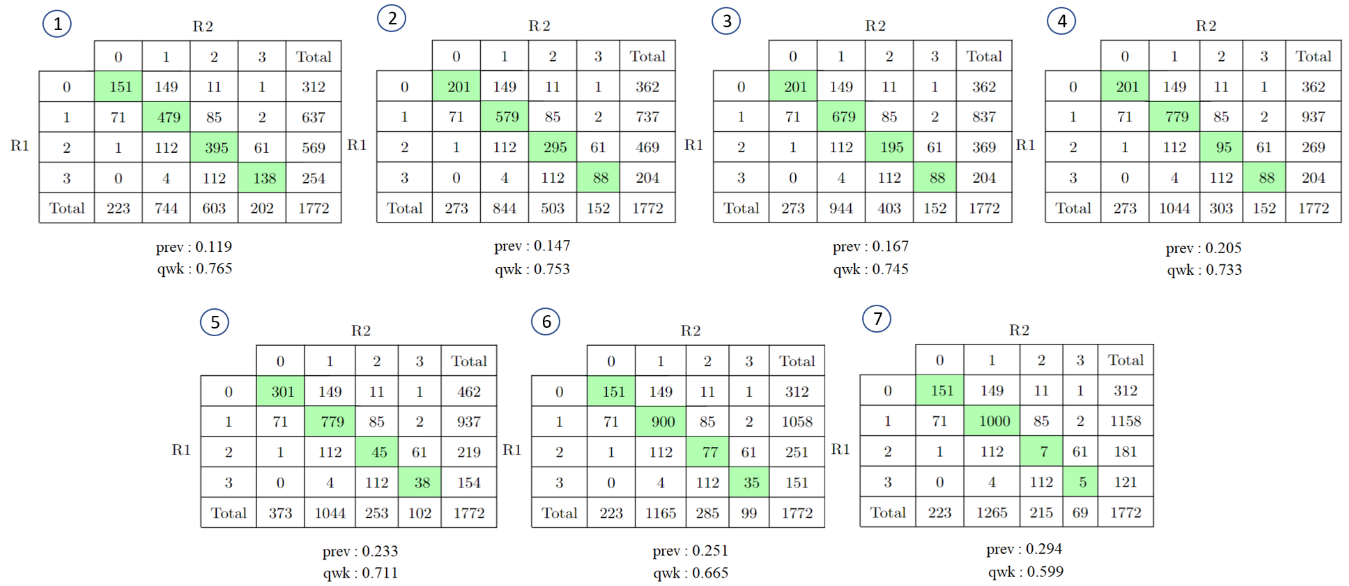
In Figure 2, two cases are presented in which there are 180 agreements and 20 disagreements between raters. In the first case, the calculated percent agreement (PI) is 0.0, and the kappa value is 0.8, while in the second case, the PI is 0.8, and the kappa value is 0.44. It is important to note that the difference between the kappa values in these two cases is due to the prevalence effect. As the value of PI increases, the expected probability (P_e) also increases, which in turn results in a decrease in the value of kappa. This relationship highlights the need to consider the prevalence of the ratings in the analysis of interrater reliability.

In essay examinations, the use of binary grades or scoring systems with only two levels is highly uncommon. Rather, the grading process typically involves multiple levels or categories of assessment. This presents a unique challenge in evaluating the agreement between human raters and automated essay scoring models, as the agreement matrix between the two will typically have a size greater than 3x3.

We propose a formula for measuring the prevalence of agreement matrix with size 3x3 or larger, as follows:

$$prev = \frac{1}{n}\frac{1}{c(c-1)/2}\sum_{i=0}^{c-1}\sum_{j=i+1}^{c} \mid U_{ii} - U_{jj} \mid \qquad (4)$$

where c is the number of classes, n is the number of items, and U_{ii} is the diagonal element of the agreement matrix. This formula is designed to provide a quantifiable measure of the average difference of all unique pairs of the categories in the raters' agreement. By dividing the sum of the absolute differences between the diagonal elements of the agreement matrix by the total number of items and the number of unique pairs of classes, we can obtain a normalized measure of the proportions of classes in the agreement.



1

		R2				
		0	1	2	3	Total
	0	151	149	11	1	312
	1	71	479	85	2	637
R1	2	1	112	395	61	569
	3	0	4	112	138	254
	Total	223	744	603	202	1772

prev : 0.119
qwk : 0.765

2

		R2				
		0	1	2	3	Total
	0	201	149	11	1	362
	1	71	579	85	2	737
R1	2	1	112	295	61	469
	3	0	4	112	88	204
	Total	273	844	503	152	1772

prev : 0.147
qwk : 0.753

3

		R2				
		0	1	2	3	Total
	0	201	149	11	1	362
	1	71	679	85	2	837
R1	2	1	112	195	61	369
	3	0	4	112	88	204
	Total	273	944	403	152	1772

prev : 0.167
qwk : 0.745

4

		R2				
		0	1	2	3	Total
	0	201	149	11	1	362
	1	71	779	85	2	937
R1	2	1	112	95	61	269
	3	0	4	112	88	204
	Total	273	1044	303	152	1772

prev : 0.205
qwk : 0.733

5

		R2				
		0	1	2	3	Total
	0	301	149	11	1	462
	1	71	779	85	2	937
R1	2	1	112	45	61	219
	3	0	4	112	38	154
	Total	373	1044	253	102	1772

prev : 0.233
qwk : 0.711

6

		R2				
		0	1	2	3	Total
	0	151	149	11	1	312
	1	71	900	85	2	1058
R1	2	1	112	77	61	251
	3	0	4	112	35	151
	Total	223	1165	285	99	1772

prev : 0.251
qwk : 0.665

7

		R2				
		0	1	2	3	Total
	0	151	149	11	1	312
	1	71	1000	85	2	1158
R1	2	1	112	7	61	181
	3	0	4	112	5	121
	Total	223	1265	215	69	1772

prev : 0.294
qwk : 0.599

Figure 3: Agreement tables with size of 4x4 from ASAP dataset Prompt 4. The first table was the original agreement table between the score predictions of Gradient Boosting and the score labels from human raters. In the next tables, the diagonal values were manipulated to increase the prevalence to examine its impact on QWK.

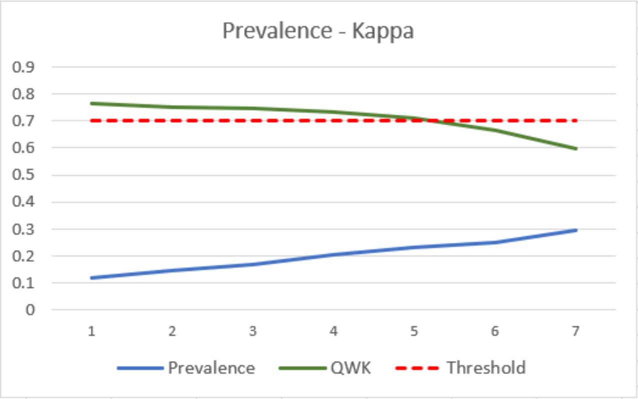


Figure 4: The prevalence's effect on QWK with respect to the acceptance threshold in AES (0.7)

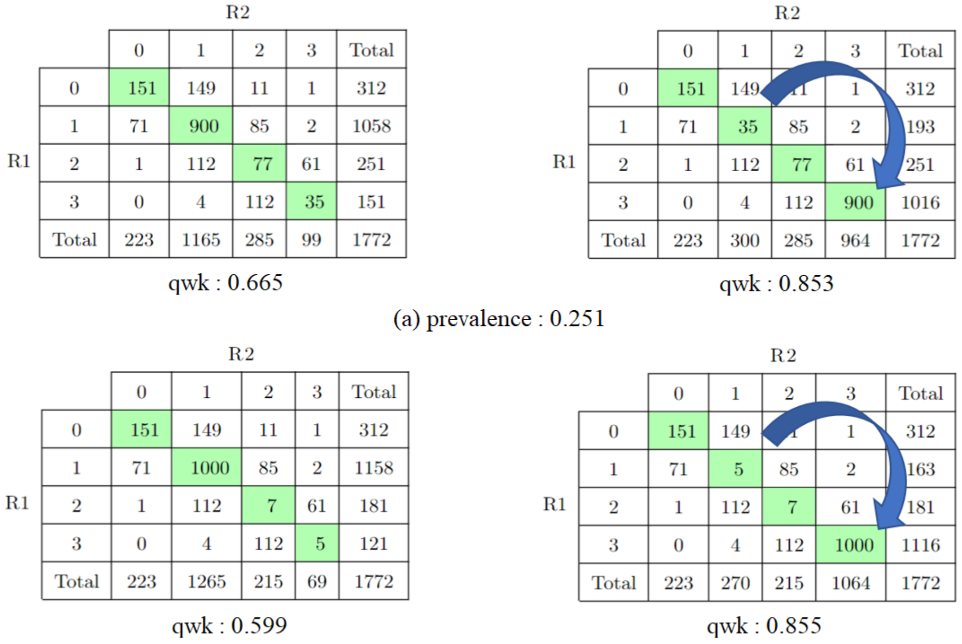
In order to further demonstrate the extent of this issue, we present an example from prompt 4 in the ASAP dataset. The scores within this dataset range from 0 to 3. In order to effectively visualize the performance of the AES model, we have included the confusion matrices that compare the model's score predictions with the human scores, which serve as the ground truth.

In order to explore the relationship between the prevalence of agreements between raters and the Quadratic Weighted Kappa (QWK) score, we conducted an experiment involving seven different proportions of agreements and visualized the results in Figure 3. The first agreement table in the figure shows the prediction performance of our trained regression model, with a QWK score of 0.765. Using our formula, the value of the prevalence is 0.119. We can observe in this table that the proportion of agreement for different scores is somewhat evenly distributed, with most of the agreement between rater 1 and rater 2 occurring in score 1. The accuracy or percent agreement for this model is 0.66.

An intriguing outcome of the QWK behavior is evidenced in the last table (no. 7) in Figure 3. It demonstrates the prediction performance with an accuracy of 0.66, which is the same as that of the first table (no. 1). However, the QWK score for this model has significantly decreased to 0.599, falling below the acceptable score of 0.70 for an AES model. Despite the decrease in the QWK score, the prevalence of agreement between the two raters in this table was found to be 0.294, indicating a higher imbalance in the agreement scores between the two raters. This scenario was created through the manipulation of the confusion matrix, in which both raters made more frequent equal predictions on score 1. This manipulation allowed for the examination of the impact of such an imbalance on the overall QWK score.

Figure 4 illustrates that as the prevalence of the agreement table increases, the QWK value decreases. Initially, the QWK value is 0.765, above the acceptance threshold, but as the prevalence increases, the QWK value drops to 0.599, below the accepted score threshold. This results in the acceptance decision for the AES model changing from accepted to rejected, even though all models have the same number of raters' agreements. These findings suggest that the proportion of agreements plays a significant role in determining the reliability of an assessment model.

The main objective of this section is to demonstrate that a scenario exists in which the value of kappa decreases as prevalence increases that leading to a decision-making challenge within the context of AES. It is worth noting, however, that this pattern of prevalence-kappa correlation is not always the case. In fact, it is possible for the kappa value



		R2				
		0	1	2	3	Total
	0	151	149	11	1	312
	1	71	900	85	2	1058
R1	2	1	112	77	61	251
	3	0	4	112	35	151
	Total	223	1165	285	99	1772

qwk : 0.665

		R2				
		0	1	2	3	Total
	0	151	149	11	1	312
	1	71	35	85	2	193
R1	2	1	112	77	61	251
	3	0	4	112	900	1016
	Total	223	300	285	964	1772

qwk : 0.853

(a) prevalence : 0.251

		R2				
		0	1	2	3	Total
	0	151	149	11	1	312
	1	71	1000	85	2	1158
R1	2	1	112	7	61	181
	3	0	4	112	5	121
	Total	223	1265	215	69	1772

qwk : 0.599

		R2				
		0	1	2	3	Total
	0	151	149	11	1	312
	1	71	5	85	2	163
R1	2	1	112	7	61	181
	3	0	4	112	1000	1116
	Total	223	270	215	1064	1772

qwk : 0.855

(b) prevalence : 0.293

Figure 5: The effect of changing the position of agreements on QWK. Both examples presented are from the last two agreement tables in Figure 3.

to increase even as prevalence increases, particularly in the context of agreement matrices with dimensions greater than 2x2. This phenomenon has not been previously addressed in the literature and will be explored in greater detail in the following section.

5.4 Position Change in Agreement

In our study, we examined the validity of the pattern proposed by Byrt et al. (1993) in relation to agreement matrices larger than 3x3. Our findings indicate that this pattern is not consistently applicable in these cases. To further demonstrate this, we analyzed two specific examples, depicted in Figure 5. Both examples presented are from the last two agreement tables in Figure 3 (tables no 6 and 7) to show an opposite relationship between prevalence and kappa, contrasting the relationship discussed in the previous section. In both cases, we maintained the same number of agreements on the diagonal of the matrix, thus preserving the overall prevalence. However, as shown in Figure 5, we observed significant changes in the Quadratic Weighted Kappa (QWK) value when altering the arrangement of these agreements within the matrix. Specifically, in the upper image of Figure 5, the QWK increased from 0.665 to 0.853 after swapping the positions of the numbers 900 and 35 in the corner of the matrix. Similarly, in the bottom image, the QWK increased from 0.599 to 0.855 after swapping the positions of the numbers 1000 and 5 on the diagonal. These findings suggest that the position of agreements within the matrix can significantly impact the QWK value and, therefore, must be considered when evaluating the agreement between raters.

The observed results can be attributed to the significant difference in the expected probability (P_e) of the two matrices being compared. As demonstrated in Figure 5(a), the matrix on the left exhibits a P_e value of 0.874, while the matrix on the right exhibits a P_e value of 0.714. It is well established that a decrease in P_e values leads to an increase in the quadratic weighted kappa (QWK) value. Similarly, the comparison presented in Figure 5(b) shows that the P_e value of the matrix on the left is 0.895, while the P_e value of the matrix on the right is 0.710. All of the cases have the same observed probability (P_o) of 0.958. These findings suggest that the P_e values of the two matrices play a critical role in determining the QWK value.

The QWK behavior in this scenario presents a challenge for decision-makers when determining whether to accept or reject an AES model. As previously discussed, the QWK scores for the agreement tables prior to the exchange of positions between two numbers are significantly lower than the minimum requirement for acceptance according to the AES model. This issue is further compounded by the fact that the kappa values for these tables shift from indicating a moderate agreement to an almost-perfect agreement. This is a significant change in interpretation despite the fact that the number of correct predictions (percent agreement) and the difference in the proportion of agreement between classes (prevalence) remain unchanged. This highlights the potentially problematic nature of relying solely on QWK scores for decision-making in regard to AES models.

5.5 The Number of Raters

In high-stakes testing programs that include writing essays among the various tasks that are measured, it is standard procedure to have multiple raters read and evaluate each of the essays, as outlined in the research of Cohen [9]. The most reliable assessment will occur when all of the responses are scored independently by different raters[2]. The greater the number of independent responses and the more the number of independent ratings of each response, the higher the reliability of the assessment will be. According to Coffman [6], the development of common examinations for English exams, rated by multiple teachers, is essential for ensuring reliability. The study suggests that utilizing two ratings, even if done quickly to allow for a larger number of ratings overall, is preferable to relying on a single rating. To further improve the reliability of rater decisions in the scoring of essays, student responses are generally scored by two or more raters, as highlighted in the research of Johnson [14]. This approach allows for a more thorough and accurate evaluation of the essays, as it takes into account multiple perspectives and ensures that any potential biases or inconsistencies are identified and addressed.

It has been noted in prior studies that there may be scenarios where more than two raters are utilized for exams grading. As exemplified in Breland's study [2], the criterion variable employed was the sum of scores obtained from four distinct essay tasks, each independently scored by four separate raters. Additionally, Johnson et al. [14] suggested that implementing three raters can also be beneficial, assuming that there is no evidence of rater drift. And it appears reasonable that the reliability of operational scores would be significantly improved by averaging the three scores from the two initial raters and the one expert.

However, one of the main limitations of using kappa statistics to assess interrater agreement is that it is only suitable for analyzing the agreement between not more than two raters. And since weighted kappa only adds weight to the observed agreement and the expected agreement matrices to the original formula of Cohen's kappa, it is also dealing with the same problem.

If we need to assess interrater agreement among a larger group of raters, we will need to use other alternatives such as Fleiss kappa [11] or Krippendorf's alpha [15]. These alternatives are specifically designed to accommodate interrater agreement metrics for more than two raters and can provide more reliable and accurate results in these situations.

Fleiss' kappa, introduced by Joseph L. Fleiss in 1971 [11], is considered an improvement over Cohen's kappa in situations where there are more than two raters or annotators involved in the assessment process. It is also noteworthy that while Cohen's kappa presumes that the same pair of raters evaluate a fixed set of items, Fleiss' kappa accommodates for variations in the composition of raters, as a fixed number of raters (e.g., three) may be assigned to varying items. Meanwhile, Krippendorff's alpha is a generalization of several known reliability indices that enables researchers to judge a variety of data with the same reliability standard. This coefficient can be applied to any number of observers, not just two, and any number of categories, scale values, or measures. Additionally, it can be used with any metric or level of measurement, including nominal, ordinal, interval, ratio, and more. Krippendorff's alpha is also suitable for handling incomplete or missing data and can be used with large and small sample sizes without requiring a minimum sample size. Overall, Krippendorff's alpha is a versatile and useful tool for assessing the reliability of different types of data.

Nevertheless, if we want to continue using kappa statistics for this specific scenario, an alternative method is to employ the calculation of pairwise averages. This approach involves determining the kappa value between rater 1 and rater 2, subsequently computing the kappa value between rater 2 and rater 3, and finally, determining the kappa value between rater 1 and rater 3. The overall inter-rater agreement is then derived by taking the mean of the three kappa agreement results. This methodology allows for a more comprehensive understanding of the agreement among raters, as it takes into account multiple pairwise comparisons. It is important to choose the appropriate metric based on the specific needs and requirements of the study in order to obtain accurate and reliable results.

6. DISCUSSION

In the preceding section, a series of experiments were conducted to thoroughly examine the behavior of Quadratic Weighted Kappa (QWK) across a range of different scenarios. Our findings demonstrate that QWK is particularly sensitive to the rating scale, with its value varying significantly in response to changes in the range of scores. The main problem is that the scores given by the two raters may be consistent, but the method used to calculate the final score can lead to inconsistencies in the acceptance of the essay scoring model. We discussed a strategy for mitigating the impact of the rating scale by changing the weights in the kappa formula. The Linear Weighted Kappa (LWK) method was found to be the most balanced method for dealing with rating scales, and it is important for decision-makers to establish a new threshold for acceptable performance criteria.

Additionally, we observed that when used in conjunction with acceptance rates of essay scoring models, the paradox of kappa can produce undesirable effects. Scoring models that perform well in terms of percent agreement or accuracy scores may not be as satisfactory when evaluated by kappa, owing to the model's inability to outperform random guessing, as the kappa statistic takes into account the possibility of agreement occurring by chance.

Furthermore, it is also crucial to consider the impact of the prevalence of an agreement matrix. Our initial experimentation yielded results that align with previous findings, as reported by Byrt et al. (1993), which suggest that as the Prevalence Index (PI) increases, the Pe value also increases, resulting in a decrease in the kappa value. This finding has significant implications for the decision-making process when evaluating the acceptance of an AES model. We developed a score prediction model for predicting scores for an essay scoring dataset (prompt 4 ASAP dataset). Despite the model's satisfactory performance in terms of the number of correct predictions, it was ultimately rejected due to a decrease in the Quadratic Weighted Kappa (QWK) value that fell below the acceptance threshold. This decline in QWK was observed as the proportion of the difference in agreement between classes increased, highlighting the importance of considering the prevalence of an agreement matrix in the evaluation of AES models.

In our study, we discovered that the correlation between prevalence and kappa for agreement tables with dimensions greater than 2x2 deviates from the pattern previously outlined in Byrt et al.'s study (1993). Specifically, we found that there is no definitive relationship between prevalence and kappa, as the behavior of kappa is highly dependent on the distribution of majority agreements within the matrix. Specifically, if the majority agreements are concentrated in the middle of the diagonal, the value of kappa will decrease, whereas if the majority agreements are located on the edges of the diagonal, the value of kappa will increase. This finding highlights the unpredictability of kappa's behavior when prevalence is held constant, and it highlights the need for caution when evaluating an AES model. Educational institutions considering the implementation of an AES system for essay score prediction should take this unpredictability into account when assessing the model's performance and determining whether to accept or reject its use.

Lastly, we must acknowledge that the use of kappa statistics is limited by the number of raters it can handle. Kappa is only suitable for assessing inter-rater agreements between up to two raters. In scenarios involving more than two raters, alternative metrics such as Krippendorf's alpha or Fleiss kappa must be employed. An alternative method for using kappa statistics in this specific scenario is to calculate pairwise averages by determining the kappa value between each pair of raters and taking the mean of the results for a more comprehensive understanding of agreement among raters. It is important to choose the appropriate metric based on the specific needs and requirements of the study in order to obtain accurate and reliable results.

The recommendation to use multiple evaluation metrics is indeed a common practice in ML. However, in the specific context of AES, we believe there is a lack of consensus on which metrics to use. Our paper provides guidance and specific recommendations for researchers and practitioners on which metrics and strategies are appropriate to mitigate different limitations of QWK in AES contexts.

7. CONCLUSION

This study examined the use of quadratic weighted kappa (QWK) as the primary evaluation metric for automated essay scoring (AES) systems. Through various experiments, we identified several limitations of QWK for its use in the context of AES, including its sensitivity to the rating scale, the occurrence of the kappa paradox, the impact of the number of agreements, and its limitation in handling a large number of raters. These characteristics of QWK can affect the acceptability of an AES system.

In summary, relying solely on QWK as the evaluation metric for AES performance may not be sufficient. It is important to consider multiple evaluation metrics when assessing the effectiveness of a model or approach. This is because different metrics can provide different insights into the performance of the model. Relying solely on one evaluation metric may not provide a complete or accurate picture of the model's performance. Additionally, using multiple evaluation metrics can increase the robustness and comprehensiveness of the evaluation, ultimately leading to more confident conclusions.

8. REFERENCES

[1] D. Boulanger and V. Kumar. Deep learning in automated essay scoring. In *International Conference on Intelligent Tutoring Systems*, pages 294–299. Springer, 2018.

[2] H. M. Breland. The direct assessment of writing skill: A measurement review. college board report no. 83-6. 1983.

[3] H. Brenner and U. Kliebsch. Dependence of weighted kappa coefficients on the number of categories. *Epidemiology*, pages 199–202, 1996.

[4] T. Byrt, J. Bishop, and J. B. Carlin. Bias, prevalence and kappa. *Journal of clinical epidemiology*, 46(5):423–429, 1993.

[5] H. Chen and B. He. Automated essay scoring by maximizing human-machine agreement. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1741–1752, 2013.

[6] W. E. Coffman. On the reliability of ratings of essay examinations in english. *Research in the Teaching of English*, 5(1):24–36, 1971.

[7] J. Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[8] J. Cohen. Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213, 1968.

[9] Y. Cohen. Estimating the intra-rater reliability of essay raters. In *Frontiers in Education*, volume 2, page 49. Frontiers Media SA, 2017.

[10] A. Doewes and M. Pechenizkiy. On the limitations of human-computer agreement in automated essay scoring. *International Educational Data Mining Society*, 2021.

[11] J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

[12] P. Graham and R. Jackson. The analysis of ordinal agreement data: beyond weighted kappa. *Journal of clinical epidemiology*, 46(9):1055–1062, 1993.

[13] R. Johnson, J. Penny, and C. Johnson. A conceptual framework for score resolution in the rating of performance assessments: The union of validity and reliability. In *annual meeting of the American Educational Research Association, New Orleans, LA*, 2000.

[14] R. L. Johnson, J. Penny, and B. Gordon. Score resolution and the interrater reliability of holistic scores in rating essays. *Written Communication*, 18(2):229–249, 2001.

[15] K. Krippendorff. Computing krippendorff's alpha-reliability. 2011.

[16] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[17] J. Liu, Y. Xu, and Y. Zhu. Automated essay scoring based on two-stage learning. *arXiv preprint arXiv:1901.07744*, 2019.

[18] R. Morris, P. MacNeela, A. Scott, P. Treacy, A. Hyde, J. O'Brien, D. Lehwaldt, A. Byrne, and J. Drennan. Ambiguities and conflicting results: the limitations of the kappa statistic in establishing the interrater

reliability of the irish nursing minimum data set for mental health: a discussion paper. *International journal of nursing studies*, 45(4):645–647, 2008.

[19] A. Sharma, A. Kabra, and R. Kapoor. Feature enhanced capsule networks for robust automatic essay scoring. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 365–380. Springer, 2021.

[20] J. Shin and M. J. Gierl. More efficient processes for creating automated essay scoring frameworks: A demonstration of two algorithms. *Language Testing*, 38(2):247–272, 2021.

[21] K. Taghipour and H. T. Ng. A neural approach to automated essay scoring. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1882–1891, 2016.

[22] Y. Wang, Z. Wei, Y. Zhou, and X.-J. Huang. Automatic essay scoring incorporating rating schema via reinforcement learning. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 791–797, 2018.

[23] M. J. Warrens. Weighted kappa is higher than cohen's kappa for tridiagonal agreement tables. *Statistical Methodology*, 8(2):268–272, 2011.

[24] M. J. Warrens. Some paradoxical results for the quadratically weighted kappa. *Psychometrika*, 77(2):315–323, 2012.

[25] D. M. Williamson, X. Xi, and F. J. Breyer. A framework for evaluation and use of automated scoring. *Educational Measurement: Issues and Practice*, 31(1):2–13, 2012.

[26] R. Yang, J. Cao, Z. Wen, Y. Wu, and X. He. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1560–1569, 2020.

[27] S. Zec, N. Soriani, R. Comoretto, and I. Baldi. Suppl-1, m5: high agreement and high prevalence: the paradox of cohen's kappa. *The open nursing journal*, 11:211, 2017.

[28] T. Zesch, M. Wojatzki, and D. Scholten-Akoun. Task-independent features for automated essay grading. In *Proceedings of the tenth workshop on innovative use of NLP for building educational applications*, pages 224–232, 2015.

How to Open Science: Debugging Reproducibility within the Educational Data Mining Conference

Aaron Haim
Worcester Polytechnic Institute
100 Institute Road
Worcester, Massachusetts
01609, USA
ahaim@wpi.edu

Robert Gyurcsan
Worcester Polytechnic Institute
100 Institute Road
Worcester, Massachusetts
01609, USA
rfgyurcsan@wpi.edu

Chris Baxter
Worcester Polytechnic Institute
100 Institute Road
Worcester, Massachusetts
01609, USA
wcbaxter@wpi.edu

Stacy T. Shaw
Worcester Polytechnic Institute
100 Institute Road
Worcester, Massachusetts
01609, USA
sshaw@wpi.edu

Neil T. Heffernan
Worcester Polytechnic Institute
100 Institute Road
Worcester, Massachusetts
01609, USA
nth@wpi.edu

ABSTRACT

Despite increased efforts to assess the adoption rates of open science and robustness of reproducibility in sub-disciplines of education technology, there is a lack of understanding of why some research is not reproducible. Prior work has taken the first step toward assessing reproducibility of research, but has assumed certain constraints which hinder its discovery. Thus, the purpose of this study was to replicate previous work on papers within the proceedings of the *International Conference on Educational Data Mining* to accurately report on which papers are reproducible and why. Specifically, we examined 208 papers, attempted to reproduce them, documented reasons for reproducibility failures, and asked authors to provide additional information needed to reproduce their study. Our results showed that out of 12 papers that were potentially reproducible, only one successfully reproduced all analyses, and another two reproduced most of the analyses. The most common failure for reproducibility was failure to mention libraries needed, followed by non-seeded randomness.

All openly accessible work can be found in an Open Science Foundation project[1].

Keywords

Open Science, Peer Survey, Reproducibility

1. INTRODUCTION

The adoption of open science and robustness of reproducibility within fields of research has incrementally gained traction over the last decade [32, 34]. This adoption trend has led to increased clarity in methodologies, easier execution of analyses, greater understanding of the underlying work, etc. However, in numerous sub-disciplines of education technology, there tends to be a lack of understanding as to why an author's research is not replicable, or even reproducible. For example, within the sub-discipline of 'Educational Data Mining', which has provided large-scale data for analyzing student learning and improve outcomes [3, 29], there are numerous analyses that, while typically falling within the reported confidence intervals, do not produce the exact results reported in the published, peer reviewed paper.

Previous works related to open science and robustness of reproducibility were conducted on the *International Conference on Learning Analytics and Knowledge* (LAK) [9, 18] and the *International Conference on Artificial Intelligence in Education* (AIED) [10]. Within the LAK work, 5% of papers were found to adopt some of the chosen practices needed for reproducibility; however, none were successful within a 15-minute timeframe. Within the AIED work, 7% of papers were reported to be 'potentially' reproducible through source analysis with some given assumptions; however, once again none were successful. The AIED work also collected responses from authors in association to their paper, in which 58% of authors reported that they could release a dataset or source needed for reproducibility; however, it did not improve the end result. These prior works only perform a basic overview of the potential reproducibility due to the given time limit, regardless of any extensions. In addition, the authors made certain assumptions that made a paper not reproducible to improve the efficiency of the reviewing process: non-defined libraries, non-seeded randomness, etc.

The goal of this work is to provide a deeper dive into the reproducibility of papers within the field of Educational Data Mining. Specifically, this work will replicate the results of previous work across papers published within the last two

[1] https://doi.org/10.17605/osf.io/unhyp

A. Haim, R. Gyurcsan, C. Baxter, S. T. Shaw, and N. T. Heffernan. How to open science: Debugging reproducibility within the educational data mining conference. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 114–124, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115651

years of the proceedings of the *International Conference on Educational Data Mining* (EDM). Trained reviewers examined each paper for open science practices and reproducibility (henceforth referred to as our peer review). We further reached out to authors in an effort to obtain more information about the paper to improve reproducibility.

Each paper was given a hard limit, with minor exceptions, of 6 hours to attempt to reproduce the paper, including communication with the authors. The process needed to attempt reproduction was recorded in a document, along with a breakdown of how much time was needed to do so. If results were obtained that did not reflect those within the paper, then an additional review of the source was conducted to determine the disconnect.

Specifically, this work aimed to accomplish the following tasks:

1. Document and analyze which papers within the proceedings of the *International Conference on Educational Data Mining* (EDM) adopt the open science practices and associated subcategories defined by this work.
2. Communicate with the authors of the papers using a survey to measure the understanding and adoption of open science practices and receive additional information to properly reproduce or replicate the paper, if needed.
3. Attempt to reproduce the paper within a 6-hour timeframe, document any additional methodologies not reported within the paper or its resources, and determine, if necessary, why the exact results reported in the paper could not be obtained.

2. BACKGROUND

2.1 Open Science

Open science is an 'umbrella' term used to describe when the methodologies, datasets, analysis, and results of any piece of research are accessible to all [15, 34]. In addition, there are subcategories of 'open science' corresponding to individual topics created before and after the initial adoption in the early 2010s [32]. Within the first half of the decade, there were numerous issues when conducting peer reviews of other researchers' work including, but not limited to, ambiguity in methodology, incorrect usage of materials, etc. Then in the mid-2010s, large-scale studies in psychology [6] and other fields [2] were unable to be reproduced or replicated. As such, open science practices were more commonly adopted to provide greater transparency and longer-lasting robustness in a standardized format such that researchers can adapt and apply their work.

Our personal investment in documenting the adoption and robustness of research in our discipline and its subfields stemmed from our own shortcomings. Specifically, our lab ran into an issue one day where we could not reproduce our prior research. There was a lack of information on how to run the analysis code, minimal information on the provided dataset, and hard-to-diagnose issues when attempting to reproduce the results. The issues were eventually solved with communication from the original author who had since left our lab, but it motivated us to do a better job at making our work more clear and more reproducible. Admitting first our own lack of adoption and ability to reproduce our work, our goals of the current work were to investigate the current adoption of open science, survey authors for their reasons for or against adoption, and attempt to reproduce their work and properly diagnose any issues that arise.

2.2 Data Mining

Data Mining is a term used to describe the extraction of previously unknown or potentially useful information from some piece of data [5, 27]. Originally known as 'Knowledge Discovery in Databases' (KDD), it has since expanded to apply the collected information in numerous fields and contexts. Within education, 'Educational Data Mining' has helped collect data on how students learn and teachers provide information at numerous levels (e.g. classroom, school, district) to better improve a student's understanding and outcomes [3, 29]. There were a few workshops in educational data mining since 2005, but in 2008, the *International Conference on Educational Data Mining* (EDM) was created [1] and took the role of hosting research which collected and analyzed large-scale data in educational settings. The collection and analysis associated with data mining practices tend to correspond with those related to open science and is typically a common topic due to developing proper and secure policies [35]. As such, papers submitted to the EDM conference will be used as the dataset for this work.

3. METHODOLOGY

3.1 Open Science Peer Review

To complete RQ1, we adopted the methodology from the previous works [9, 10]. We evaluated every full paper, short paper, and poster paper from the previous two EDM proceedings: the *15th International Conference on Educational Data Mining*[2] and the *14th International Conference on Educational Data Mining*[3]. Reproducibility of older years was likely to be more difficult as papers become older as software might no longer exist or is outdated or the dataset or source required had been taken down for some reason. Thus, only the last two years were considered. Both proceedings are divided into subsections 'Full Papers' (synonymous with research articles in previous works), 'Short Papers', or 'Poster Papers' (synonymous with posters in previous works). The papers within the proceedings of the *15th International Conference on Educational Data Mining* were identified by their digital object identifier (DOI)[4]. The papers within the proceedings of the *14th International Conference on Educational Data Mining* were identified by their page number within the proceedings[5]. As the identifiers for each proceeding were different but functionally equivalent, they were referred to as unique identifiers (UID). Each review captured a UID, the

[2] https://zenodo.org/communities/edm-2022/

[3] https://educationaldatamining.org/EDM2021/EDM2021Proceedings.pd

[4] There was no DOI associated with the proceedings itself, so the citation is a footnote with a link to the community group on Zenodo.

[5] The proceedings of the *14th International Conference on Educational Data Mining* had no DOI. As such, the page number in the proceedings were used. A separate link was provided to the virtual page for each paper as well.

proceedings the paper was a part of, and the subsection the paper was listed under. Each review for a paper was given a maximum time limit of 15 minutes because of logistical constraints (e.g. non-specified or degraded links, nested resources within citations, etc.). In addition, an explanations document was created which justified why a specific choice was made in the review. If a choice was self-explanatory, the justification was omitted (e.g., no preregistration was linked in the paper, no README was located in the source). Any links within the paper that no longer reference the original resource were marked as degraded and reported in the explanations document.

Open Methodology is a term that says the details of the collection, methods, and evaluation of a research project are accessible and usable by all [15]. Compared to a paper, the methodologies typically represent every possible step and resource needed for another researcher to reproduce or replicate the research themselves. All papers submitted to the *15th International Conference on Educational Data Mining* are licensed under the Creative Commons Attribution 4.0 International License[6], or CC-BY-4.0 for short, and are considered 'Open Access'. The papers within the proceedings of *14th International Conference on Educational Data Mining* are unlicensed; however, EDM treats them as 'Open Access' regardless, so they are considered as such for this work.

Open Data is a term that says the dataset(s) associated with the research project is accessible and can be used by all [17, 19]. These datasets are typically specified with a license or are part of the public domain. A dataset is marked as being open if the paper contains a link, or a link to another paper with a link, to the dataset. If the paper mentions explicitly that the dataset can be requested from the authors, then it will be marked as 'on request'. If the paper does not use a dataset, such as for theoretical or development topics, then the field is marked as non-applicable. The licensing on the dataset was not considered as researchers are unlikely to be as familiar with them and are normally ambiguous or too complex to properly understand [13, 28]. A separate field is provided for the documentation of the data which is marked if there exists a location where the dataset's fields are mapped to its associated description. A partial marking for the documentation can be met if there is at least one field documented at some location.

Open Materials is a term that captures whether technologies – including open source software [25, 11], freeware, or non-restrictive services – can be used by all. A paper has open materials if the paper contains a link, or a link to another paper with a link, to all the materials and source the authors used. A partial marking was assigned if there is at least one material mentioned. If there are no materials used, such as for argumentative or theoretical papers, then the field was marked as non-applicable. The documentation for the materials and source, which provides understanding on how to use them [7], also had a field, along with a partial equivalent if the materials or source was not fully documented. If the source was available, then two more fields were considered: the README which contained information on the source and potentially some setup instructions [14] and a license field which said that the source can be used openly [25, 31, 8].

A **preregistration** describes the processes conducted for the paper before the research takes place to prevent hypothesizing after results are known and p-hacking observations [22, 21, 33]. Preregistrations can range in complexity, from documenting *a priori* sample sizes, to exclusion criteria, to full analysis plans. While they are often believed to solely be used for null hypothesis testing, preregistrations can be, and are used in a wide-range of research methodologies including qualitative methods and secondary data analysis. A preregistration can be altered by creating a new preregistration to preserve the initial methodologies. A paper has a preregistration if there is a link within the paper to some location hosting the preregistration (e.g., Open Science Framework[7], AsPredicted[8]). If a preregistration is unnecessary, then the field is marked as non-applicable.

In contrast to previous works, the peer review was handled by two trained undergraduate research assistants, referred to as 'Reviewers' in the explanations document. Undergraduates are typically pressed upon to conduct and publish research prior to graduation for better advancement within their career [16, 30, 26]. As such, it stands to reason that papers should be geared towards the understanding of undergraduates assuming the requisite knowledge. Due to undergraduate interpretation, it was expected to see a higher level of adoption as previous works tended to be highly specific and nuanced when evaluating whether a given subcategory was adopted.

To mitigate any misconceptions or inaccuracies between the reviewers, each reviewer was randomly assigned ten papers that another reviewer reviewed and provided their own review. Both reviews are provided within the explanations document in an arbitrary order.

As a final precaution, the lead on the research project, referred to as the 'Meta-Reviewer', was responsible for resolving any disputes or disagreements within the provided reviews. If two reviewers disagreed on a particular section, the meta-reviewer had the final say as to what was reported. Additionally, if either reviewer asked for verification on a particular review, the meta-reviewer provided the requested feedback and correct markings. Finally, the meta-reviewer lightly reviewed the results of the reviewers for any major inaccuracies in understanding or logic and corrected them as necessary.

3.2 Author Survey

To complete RQ2, Authors were allowed to provide input to the peer review performed using a survey. For each paper submitted to the two EDM conferences, an email was sent out to the first author[9]. To avoid issues involving the email server (e.g. email marked as spam, denied due to too many receipts), authors with multiple papers published in the proceedings were sent a single email containing the papers they

[6] https://creativecommons.org/licenses/by/4.0/

[7] https://osf.io/registries

[8] https://aspredicted.org/

[9] The first author was assumed to be the corresponding author as EDM does not provide any formal way of marking so.

should complete the survey for[10]. As an added measure to improve the number of survey responses, a separate, mass email was sent prior to the survey to notify authors about the survey and what email it would be sent from. The survey responses were publicly released and linked by their UI as stated in our International Review Board (IRB) study. Additionally, the author information provided was removed from the released dataset. The survey itself was sent on November 29th, 2022 and currently continues to collect responses. This work reports on responses collected up to January 3rd, 2023.

The survey asked for the name and email of the author and the UI of the associated paper. The content of the survey was separated into six subsections: data, materials, preregistration, preprint, reproducibility and replicability, and resource degradation.

3.2.1 Data

The data section was used to collect information on the dataset and documentation used within the paper. The author first reported whether the dataset is publicly available, is private but can be shared on request, or if the dataset cannot be shared at all. In the case where a dataset was not used or does not correspond with one of the above categories, an additional 'other' option was available with an appropriate text box. If the dataset was not publicly accessible, the author was asked to provide their reasoning as to why. If the dataset could be shared either publicly or on request, the author was asked to provide the location of the dataset along with its associated license. If a link was provided but the dataset could not be released publicly, the link would be scrubbed from the publicly released dataset. This would provide a relatively secure way to share data that may contain sensitive information. All questions were shown for full transparency.

3.2.2 Materials

The materials section was responsible for collecting information on the materials, source, and documentation used within the paper. The questions in this section are the same as those within the data section except replaced with material-related keywords.

3.2.3 Preregistration

The preregistration section was responsible for collecting information on an available preregistration, if applicable, for the paper. The author was asked to report on whether there is a public, private, or no preregistration made for the paper. If a preregistration was not applicable (e.g. theoretical paper, argumentative paper) or did not fit into one of the available categories, an additional 'other' option was available with an appropriate text box. For available preregistrations, whether public or private, the author was requested to provide the associated link. If no preregistration was made, the author was asked to provide their reasoning as to why.

3.2.4 Preprint

The preprint section documented information on an available **preprint**, a paper that usually proceeds formal peer review and publication in a conference or journal [4, 12], for the paper. The author was asked to report on whether a preprint was available for the paper. If a preprint was not applicable or did not fit into one of the available categories, an additional 'other' option was available with an appropriate text box. If a preprint was present, the author was requested to provide the associated link. If no preprint was created, the author was asked to provide their reasoning as to why.

3.2.5 Reproducibility and Replicability

The reproducibility and replicability section documented information needed to properly reproduce or potentially replicate the associated paper. Towards replication, the author was asked to provide any additional methodologies that were not reported in the original paper. Towards reproduction, the author was asked to provide any necessary setup instructions needed to properly connect the dataset to the source and run the associated analysis. This included, but was not limited to, file locations, software versions, setup scripts, etc. If any of the above information was not provided within the paper or its citations, the author was asked to provide their reasoning as to why.

3.2.6 Resource Degradation

The resource degradation section documented information on resources reported within the papers that no longer exist at the specified location. The authors were asked to review their resources for any that no longer exist or point to an incorrect location and provide alternatives if possible. If the resources were degraded, the author was asked to explain what happened to the original resource.

3.3 Reproducibility

An experiment or study is **reproducible** when the exact results reported in the paper can be produced from a static input (e.g. dataset, configuration file) and deterministic methodology (e.g. source code, software)[20, 24, 23]. While reproducibility is the simplest form of reviewing the results of a paper, in practice, there are differing levels of what defines a complete reproduction. For this work to complete RQ3, we assume that a paper is reproducible when the dataset and analysis used in the original paper returns the exact same results and figures as those reported. If either the dataset or analysis method is not present, found within the 15-minute timeframe in the paper or its resources, or provided within the author's survey response, then the paper will be marked as non-reproducible. If the paper does not use a dataset or analysis method or does not run an experiment or study in general, then reproducibility will be marked as non-applicable.

Although we allocated 15 minutes for each paper to find its dataset or analysis, if we were able to track these down, each paper was given a hard limit of 6 hours to reproduce the results reported in the paper. If any action exceeded the 6 hour limit, then the action was stopped and only the exported results were considered with any reasonable educated guesses on the rest of the runtime. The 6 hour limit was only extended if the reproduction could be assumed to be completed

[10] This email survey was conducted in parallel with two separate research projects for other conferences to mitigate the issues mentioned above. The other research projects will be reported at a later time.

within an additional hour. To provide a better and more accurate understanding of the amount of time taken, the collected metric was broken down into three time periods: setup, execution, and debugging. A timing site[11] was used to manually track how long each section took along with the total time. If any breaks were taken by the reviewer, the timers and all actions were stopped and recorded in the explanations document until the reviewer resumed working.

The **setup time** tracked the time taken for all tasks prior to the first execution of the analysis. This includes downloading the dataset and source, setting up the necessary environment, and following information provided within the README, if available. Information that can be assumed from the source was not provided during the setup phase to better simulate cases where a researcher would run the source assuming they had all the necessary libraries installed from previous runs. This time was likely to vary between reviewers depending on factors such as connection speed and should be taken with a grain of salt. The **execution time** tracked the time taken during the execution of the program. This began when the program was ran (e.g. command, button) and stopped when the program finished executing or crashed. This time was the total time on execution any might included multiple runs. Any specific information was recorded in the explanations document. The **debugging time** tracked the time taken between executions when the analysis crashed. Any diagnoses made which corrected the issue was reported in the explanations document. A perfectly reproducible analysis should have minimal to no debugging time.

All reproducibility tests were run on a single big data machine used within the author's lab. The machine was chosen for two reasons. First, as a big data machine, it can run numerous calculations relatively quickly depending on the efficiency of the analysis. Second, it runs a Unix-based operating system with a Bash shell which most scripts provided by researchers are typically for. For benchmarking purposes, the specifications of the machine are listed in Appendix B.

3.3.1 Python

If the environment needed to reproduce the source used Python[12], then the following steps were taken:

1. If a specific version of Python was specified, download and select the version of Python.
2. Create a empty virtual environment using 'venv'[13] and activate it.
3. Follow any setup steps specified by the analysis.
4. If the analysis is in a Python (.py) file:
   (a) Run the file using the 'python' command.
5. If the analysis is in a Python Notebook (.ipynb):
   (a) Install 'ipykernel' and 'notebook' using the 'pip' command.[14]
   (b) Open the notebook and specify the kernel used as the one within the virtual environment.
   (c) Run the notebook.

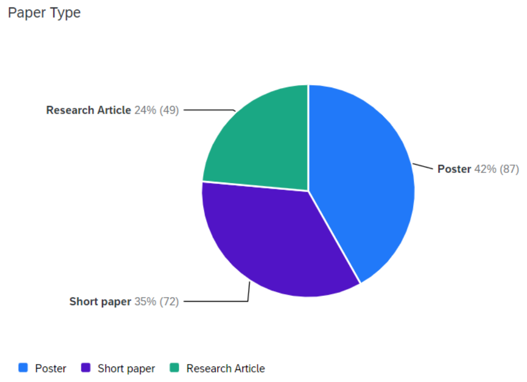


Figure 1: A representation of the review on the full papers, short papers, and poster papers published within the proceedings of the 15th and 14th EDM conferences.

3.3.2 R

If the environment needed to reproduce the source used R[15], then the following steps were taken:

1. If a specific version of R was specified, download and select the version of R.
2. Create a new project using RStudio[16] or another IDE that can use 'packrat'[17][18].
3. Follow any setup steps specified by the analysis.
4. Run the R script.

4. RESULTS

4.1 Peer Review

As shown in Figure 1, across the 99 papers published in the 15th proceedings and the 109 published in the 14th proceedings, there were 49 full papers (research articles), 72 short papers, and 87 poster papers.

As shown in Figure 2, 32, or 15%, of papers used a dataset that was already or made openly available. 5% mentioned that the dataset could be requested. Out of those 15% with openly available data, 69% had full documentation on the dataset while the other 31% had partial documentation.

[11] https://stopwatch.online-timers.com/multiple-stopwatches

[12] https://www.python.org/

[13] This is the recommended way for Python 3; however, there are other methods to do so.

[14] If the path is improperly configured, the command may need to be prefixed with 'python -m'.

[15] https://cran.r-project.org/

[16] https://posit.co/products/open-source/rstudio/

[17] https://cran.r-project.org/package=packrat

[18] 'packrat' is the most commonly used option for managing R dependencies. It is not the only method.

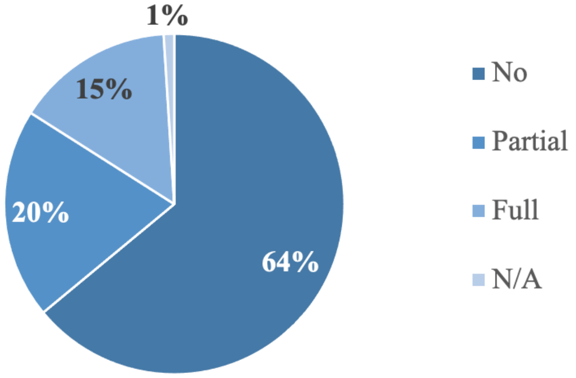


Figure 2: A representation of the review on the adoption of open data within papers published in the proceedings of the 15th and 14th EDM conferences.

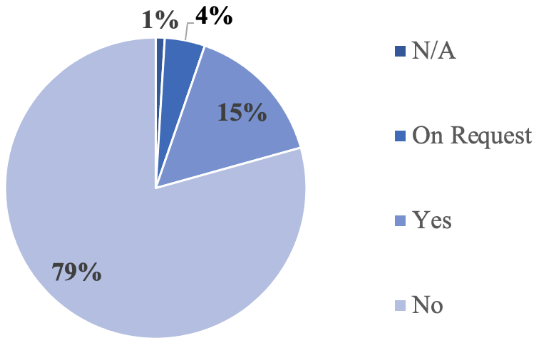


Figure 3: A representation of the review on the adoption of open materials within papers published in the proceedings of the 15th and 14th EDM conferences.

As shown in Figure 3, 31, or 15%, of papers used materials and made the source openly available. 20% used at least on openly available materials. Out of those 15% with openly available materials, 45% had full documentation while 55% had partial documentation. Additionally, 94% of the open materials had a README while 44% had a permissive license provided with the source.

As shown in Table 1, only three, or 1%, of the papers had a preregistration linked to it. One of the papers was a short paper while the remaining two were poster papers. One paper was determined to be non-applicable for having a preregistration as it was a concept discussion.

Finally, as shown in Table 2, nine, or 4%, of the papers provided dataset links that were no longer located in its original location. Two were full papers, five were short papers, and the remaining two were poster papers. Six, or 3%, provided material links that were no longer available. One was a full paper, four were short papers, and the remaining one was a poster paper.

4.2 Author Survey

Out of the 208 surveys sent, only 13, or 6% of the articles, provided a complete response within the one month period. Fourteen, or 7% of the surveys, did not reach their destination in a timely fashion: two received auto response emails about a delay in reading the email, two were denied by the mail server, and ten emails were no longer available or locatable on the mail server.

Out of thirteen responses, three papers reported that their datasets were publicly available, five papers reported that their dataset could be requested, and five papers reported that they cannot share their datasets. Out of the eight public and on request responses, five did not mention in the paper that they could share or request the dataset. Out of the ten on request and cannot share responses, six mentioned they do not have the rights or necessary license to release the dataset, three mentioned that the dataset contains sensitive information due to an IRB or some other committee, and one mentioned they simply did not have enough time to go through the process of reviewing and potentially publicly releasing a dataset.

For materials, nine reported that they could make their materials and source public, Three reported that they could share their materials and source on request, and one mentioned that they cannot release their materials and source. Out of the twelve public and on request responses, eight did not mention in the paper that they could share or request the materials or source. Out of the four on request and cannot share responses, three mentioned that the source contains references to sensitive information from the associated dataset while one mentioned they simply did not have the time nor motivation to go through the process of reviewing and potentially publicly releasing their materials and source.

Towards reproducibility, only one mentioned additional information was necessary to reproduce their work while two mentioned that the information on the source should be enough to do so. The provided resources did not have an effect on the reproducibility of the papers within Section 4.3.

Table 1: A representation of the review on the adoption of preregistrations within papers published in the proceedings of the 15th and 14th EDM conferences split by paper type.

	Total	Yes (%)	No (%)	N\A (%)
Full Paper	49	0 (0%)	49 (100%)	0 (0%)
Short Paper	72	1 (1%)	71 (99%)	0 (0%)
Poster	87	2 (2%)	84 (97%)	1 (1%)

Table 2: A representation of the review on the degradation of resources within papers published in the proceedings of the 15th and 14th EDM conferences split by paper type.

	Data	Materials	Methodology	Preregistration
Full Paper	2	1	0	0
Short Paper	5	4	0	0
Poster	2	1	0	0

One survey response mentioned that they did create a preregistration and provided a link to it while twelve did not. Out of the twelve who did not create a preregistration, four believed that one was not necessary during the beginning of the research project, one did not remember the option existed, and six did not know what a preregistration was. One provided no response.

Five survey responses reported that they did create a preprint while eight did not. Out of the five that created a preprint, only four links were provided. Out of the eight that did not create a preprint, two believed that one was not necessary, two did not remember the option existed, two did not know what a preprint was, and one did not believe it was fair to the review process to create a preprint. One provided no response.

No survey responses reported anything about their resources no longer existing at the specified location.

4.3 Reproducibility

Only twelve, or 6% of papers, were able to attempt reproduction. Two papers were unable to be timed due to logistical reasons during setup. One paper requested a Python dependency which was no longer obtainable in an official capacity. The other paper required arguments to run the Python script which were not defaulted. There was no indication as to what the value of those arguments might be, so there was an infinite number of potential combinations. As such, the paper was deemed to be non-reproducible.

Five papers passed the 6-hour hard limit. One paper was excused because of the additional overflow, but it did not allow all the results to be completed. Two papers were still running during the execution time when the 6-hour limit passed; however, only one produced intermediate results that could be compared. One paper crashed 30 minutes before the limit and provided intermediate results. The remaining paper was being debugged as there were a number of errors and version incompatibilities between the Python libraries preventing execution which was specific configured.

Only nine of the ten tested papers required some amount of debugging. The remaining paper, while needing no debugging, produced numerous results that did not line up with those reported. Out of the nine papers which required debugging, all nine were missing some unreported dependency that needed to be downloaded. Two papers failed as their source code did not create the necessary directories to read or write files to.

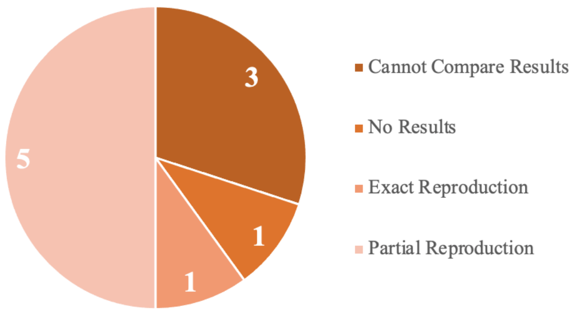


Figure 4: A representation of the test results obtained while reproducing papers published in the proceedings of the 15th and 14th EDM conferences.

As shown in Figure 4, out of the ten tested papers, only six produced results that could be potentially linked to the paper or its resources. Three papers provided results but not in a comparable form to the paper. The remaining paper passed the 6-hour time limit due to version incompatibilities. Only one paper, a poster paper, exactly provided the results expressed within the paper; however, some of the results had to be pulled from an intermediate variable that was not printed. The remaining five provided some of the results reported in the paper; however, only two papers could safely mitigate the inaccuracies due to the confidence interval.

Further source analysis revealed the five papers which did not exactly provide the results mentioned in the paper was due to non-seeded randomness: the seed, or initial value, which in most cases makes the numbers generated by the algorithm fixed instead of random is not set to a deterministic value. Some papers do partially set the seed for some generators but not all.

5. LIMITATIONS AND FUTURE WORK

A number of limitations within previous works replicated for RQ1 and RQ2 are still applicable to this paper due to human intervention and limited resources [9, 10]. For the peer review, this includes the subjectiveness of the author's review on the proceedings papers and mitigation through an explanations document. For the author survey, this includes the nonexistent fallback strategy, confusion of email and survey instructions, and the limited responses. As such, any conclusions made would only accurately reflect a subset of the educational data mining community.

As the peer reviews were conducted by undergraduate research assistants, there are likely some misconceptions between the instructions given, the understanding of the papers, and the explanations for their choices. To better standardize and mitigate these concerns, the undergraduates were each given a standardized set of explanations which could be used during review. In addition, examples were given to better understand the relationship between the review topic and its corresponding phrases within the papers and associated resources. As an added precaution, the undergraduates could ask a graduate student to perform a meta-review or review other undergraduates' reviews in either agreement or disagreement.

We did not conform to a single framework to measure the reproducibility of a paper. This was because research, along with its resources, are not uniform in implementation. The papers we attempted to reproduce in this study used a wide variety of services, data, and materials stored across numerous locations: GitHub, personal websites, the Open Science Framework, direct downloads, etc. In addition, each paper had differing levels of documentation, licensing, interoperability, and replicability. We mitigated this by using broad categories and definitions with delineated cutoffs when defining our methodology. However, future work might want to review how well the papers meet existing frameworks, such as FAIR[19].

When testing for reproducibility, the total time spent had a hard limit, with one exception, of 6 hours. 2 of the available papers were halted due to this limit; however, only one did not produce intermediate results that could be compared to a paper. It would be useful to properly test the execution for the entire time provided a large number of machines were available.

Additionally, the timing was performed manually instead of through timers associated with the application. There could be slight overestimations in the amount of time taken to reproduce. On the other hand, software timers are ill-suited for such a task as they are typically not multilingual and may not be available for all software.

Future work should include another round of reproducibility tests on different machines. Each test would provide a valid benchmark on the execution length of the code and serve as a robust measure to validate the reproducibility in numerous circumstances. In addition, results that were inaccurate due to randomness could be averaged to provide a more accurate estimate of the results compared to those reported. Authors could be recruited to run reproducibility tests either voluntarily or through giveaways; however, it would require the authors to have a greater understanding of computer science rather than those needed to provide their analyses.

Another direction for future work could view the impact of conferences which promote open science and reproducibility measures to compare them to those without them. In addition to previous work on author responses and this work on reproducibility, a comparison could be made between the promoting and non-promoting conferences to see whether the adoption of such practices have improved the robustness of research within the discipline.

Finally, the timer categories could be more specified and less generalized. Each timer only represents the length of each section rather than individual sections for how long a specific task took. For the setup and debug categories, these specific sections would not be as useful since different reviewers might take different lengths of time to setup or determine an issue. For the execution category, while it would be useful to know how much time was needed to reproduce the results, it would be better suited as a benchmark from the original author who had already ran the methodology successfully and without issue.

6. CONCLUSION

Approximately 35% of papers met a partial definition of the chosen open science practices with 5% able to attempt reproducibility with the combined peer review and author survey responses. With the additional time compared to previous works, one paper provided the exact results reported in the paper, while two papers mostly provided the reported results. In addition, while all of the papers needed to download unreported libraries to properly execute the source, the non-exact results collected could all be attribute to non-seeded randomness.

In-depth reproducibility tests and source analysis greatly increases the robustness of an author's paper. The two main issues within the paper might not seem relevant to most authors, but they are likely to have some lasting impact in the future. Library compatibility may not seem useful in a year or two, but after half a decade or so, trying to run the same analysis might prove to be impossible as it did with two of the papers in this work. As for non-seeded randomness, most researcher would agree that as long as the obtained value is within the confidence interval, then it should considered replicable. However, a lack of stability across papers might lead to one reproduction compared to another reproduction, which is not guaranteed to be within each other's confidence interval. As such, deterministic results provide greater robustness and stability such that it can stand the test of time.

Most of the issues can be simplified down to a few additional actions necessary to provide deterministic results. Taking Python analyses as an example, the libraries could be exported with the source by running the 'pip freeze' command. Any source of randomness within Python or popular libraries can also be seeded such as 'random.seed' or, for

[19] https://www.go-fair.org/fair-principles/

numpy[20], 'numpy.random.seed'. Other languages or sources are not much different. In the cases where libraries are no longer present, the container itself can be wrapped and provided using services like containerd[21]. By providing these simple, quick actions, the robustness of research, and open science in general, could be greatly improved.

7. ACKNOWLEDGMENTS

We would like to thank the NSF (e.g., 2118725, 2118904, 1950683, 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, & 1535428), IES (e.g., R305N210049, R305D210031, R305A170137, R305A170243, R305A180401, R305A120125, & R305R220012), GAANN (e.g., P200A180088 & P200A150306), EIR (U411B190024 & S411B210024), ONR (N00014-18-1-2768), and Schmidt Futures. None of the opinions expressed here are that of the funders. We are funded under an NHI grant (R44GM146483) with Teachly as a SBIR. We would also like to extend our thanks to the blinded reviewers who provided their suggestions towards improving this paper.

8. REFERENCES

[1] M. Al-Razgan, A. S. Al-Khalifa, and H. S. Al-Khalifa. Educational data mining: A systematic review of the published literature 2006-2013. In T. Herawan, M. M. Deris, and J. Abawajy, editors, *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, pages 711–719, Singapore, 2014. Springer Singapore.

[2] M. Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454, May 2016.

[3] R. Baker et al. Data mining for education. *International encyclopedia of education*, 7(3):112–118, 2010.

[4] P. E. Bourne, J. K. Polka, R. D. Vale, and R. Kiley. Ten simple rules to consider regarding preprint submission. *PLOS Computational Biology*, 13(5):1–6, 05 2017.

[5] M.-S. Chen, J. Han, and P. Yu. Data mining: an overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):866–883, 1996.

[6] O. S. Collaboration. Estimating the reproducibility of psychological science. *Science*, 349(6251):aac4716, 2015.

[7] B. Dagenais and M. P. Robillard. Creating and evolving developer documentation: Understanding the decisions of open source contributors. In *Proceedings of the Eighteenth ACM SIGSOFT International Symposium on Foundations of Software Engineering*, FSE '10, page 127–136, New York, NY, USA, 2010. Association for Computing Machinery.

[8] A. Engelfriet. Choosing an open source license. *IEEE software*, 27(1):48–49, 2009.

[9] A. Haim, S. Shaw, and N. Heffernan. How to open science: A principle and reproducibility review of the learning analytics and knowledge conference. In *LAK23: 13th International Learning Analytics and Knowledge Conference*, LAK2023, page 156–164, New York, NY, USA, 2023. Association for Computing Machinery.

[10] A. Haim, S. T. Shaw, and I. Heffernan, Neil T. How to open science: A reproducibility author survey of the artificial intelligence in education conference, Apr 2023.

[11] J. Johnson-Eilola. Open source basics: Definitions, models, and questions. In *Proceedings of the 20th Annual International Conference on Computer Documentation*, SIGDOC '02, page 79–83, New York, NY, USA, 2002. Association for Computing Machinery.

[12] J. Kaiser. The preprint dilemma. *Science*, 357(6358):1344–1349, 2017.

[13] M. Khayyat and F. Bannister. Open data licensing: more than meets the eye. *Information Polity*, 20(4):231–252, 2015.

[14] M. Koskela, I. Simola, and K. Stefanidis. Open source software recommendations using github. In E. Méndez, F. Crestani, C. Ribeiro, G. David, and J. C. Lopes, editors, *Digital Libraries for Open Knowledge*, pages 279–285, Cham, 2018. Springer International Publishing.

[15] P. Kraker, D. Leony, W. Reinhardt, and G. Beham. The case for an open science in technology enhanced learning. *International Journal of Technology Enhanced Learning*, 3(6):643–654, 2011.

[16] M. C. Linn, E. Palmer, A. Baranger, E. Gerard, and E. Stone. Undergraduate research experiences: Impacts and opportunities. *Science*, 347(6222):1261757, 2015.

[17] J. C. Molloy. The open knowledge foundation: Open data means better science. *PLOS Biology*, 9(12):1–4, 12 2011.

[18] B. Motz, C. Brooks, J. Quick, Y. Bergner, G. Gray, C. Lang, W. Li, and F. Marmolejo-Ramos. A baseline measure of open research practices in learning analytics, Mar 2022.

[19] P. Murray-Rust. Open data in science. *Nature Precedings*, 1(1):1, Jan 2008.

[20] E. National Academies of Sciences, P. Affairs, E. Committee on Science, B. Information, D. Sciences, C. Statistics, B. Analytics, D. Studies, N. Board, D. Education, et al. *Reproducibility and Replicability in Science*. National Academies Press, Washington, D.C., USA, 2019.

[21] B. A. Nosek, E. D. Beck, L. Campbell, J. K. Flake, T. E. Hardwicke, D. T. Mellor, A. E. van 't Veer, and S. Vazire. Preregistration is hard, and worthwhile. *Trends in Cognitive Sciences*, 23(10):815–818, Oct 2019.

[22] B. A. Nosek, C. R. Ebersole, A. C. DeHaven, and D. T. Mellor. The preregistration revolution. *Proceedings of the National Academy of Sciences*, 115(11):2600–2606, 2018.

[23] B. A. Nosek, T. E. Hardwicke, H. Moshontz, A. Allard, K. S. Corker, A. Dreber, F. Fidler, J. Hilgard, M. Kline Struhl, M. B. Nuijten, J. M. Rohrer, F. Romero, A. M. Scheel, L. D. Scherer, F. D. Schönbrodt, and S. Vazire. Replicability, robustness, and reproducibility in psychological science. *Annual Review of Psychology*, 73(1):719–748, 2022. PMID:

[20] https://numpy.org/

[21] https://containerd.io/

34665669.

[24] P. Patil, R. D. Peng, and J. T. Leek. A statistical definition for reproducibility and replicability. *bioRxiv*, 1(1):1–1, 2016.

[25] B. Perens et al. The open source definition. *Open sources: voices from the open source revolution*, 1:171–188, 1999.

[26] J. K. Petrella and A. P. Jung. Undergraduate research: Importance, benefits, and challenges. *International journal of exercise science*, 1(3):91, 2008.

[27] G. Piateski and W. Frawley. *Knowledge Discovery in Databases*. MIT Press, Cambridge, MA, USA, 1991.

[28] G. K. Rajbahadur, E. Tuck, L. Zi, Z. Wei, D. Lin, B. Chen, Z. M. Jiang, and D. M. German. Can I use this publicly available dataset to build commercial AI software? most likely not. *CoRR*, abs/2111.02374:1–1, 2021.

[29] C. Romero and S. Ventura. Data mining in education. *WIREs Data Mining and Knowledge Discovery*, 3(1):12–27, 2013.

[30] S. H. Russell, M. P. Hancock, and J. McCullough. Benefits of undergraduate research experiences. *Science*, 316(5824):548–549, 2007.

[31] H. Schoettle. Open source license compliance-why and how? *Computer*, 52(08):63–67, aug 2019.

[32] B. A. Spellman. A short (personal) future history of revolution 2.0. *Perspectives on Psychological Science*, 10(6):886–899, 2015. PMID: 26581743.

[33] A. E. van 't Veer and R. Giner-Sorolla. Pre-registration in social psychology—a discussion and suggested template. *Journal of Experimental Social Psychology*, 67:2–12, 2016. Special Issue: Confirmatory.

[34] R. Vicente-Saez and C. Martinez-Fuentes. Open science now: A systematic literature review for an integrated definition. *Journal of Business Research*, 88:428–436, 2018.

[35] A. Zuiderwijk and M. Janssen. Open data policies, their implementation and impact: A framework for comparison. *Government Information Quarterly*, 31(1):17–29, 2014.

APPENDIX

A. STANDARD PHRASES

This was a list of standard phrases used within the explanations document which was used to provide information or justifications on a given paper. The text might have been changed or further elaborated when used:

- The raw dataset and materials do not seem to be provided anywhere.
  - This was used when there is no information or links provided on the dataset or materials within the paper or its sub-resources. This might have also been used if it took longer than 15 minutes to located the associated resource(s).
- The raw dataset does not seem to be provided anywhere.
  - This was used when there is no information or links provided on the dataset within the paper or its sub-resources. This might have also been used if it took longer than 15 minutes to located the associated resource(s).
- The data documentation is likewise nonexistent.
  - This was used when there was no information within the paper on any documentation of the columns of the dataset. This was typically used in conjunction with papers that did not provide the dataset.
- Some data documentation is represent through <location>, and as such it will be marked as partial.
  - This was used when a column within the dataset was found to be marked in a paper or its sub-resources. The 'location' was replaced with the section or link the description was located.
- Open Materials include <materials>.
  - This was used whenever a paper contained materials that were not mentioned in the source or that the source was not provided for in the paper. The 'materials' was replaced with a list of the materials and links to their locations, if possible.
- The full analysis is not provided, so the materials fields will be marked as partial.
  - This was used when the source was unavailable when materials were present, or when the source did not seem to provide the ability to replicate all results provided within the paper.
- The paper seems to be argumentative in nature to create a new theoretical idea to use in the field. As such, all of the fields will be marked as non-applicable.
  - This was used when a paper talked about or elaborated on a concept rather than conduct an experiment or study. It marked all the available open science topics as non-applicable.

B. COMPUTER SPECIFICATIONS

B.1 Hardware Components

- AMD Ryzen Threadripper 2950X[22]
- NVIDIA GeForce RTX 3090[23]
- Corsair VENGEANCE LPX 128GB (4 x 32GB) DDR4 DRAM 2133MHz C18 Memory Kit
- WD Blue SN550 NVMe SSD (WDC WDS200T2B0C-00PXH0)[24]

B.2 Software Components

Some of the software components are considered the default if no specific version was specified in Section 3.3.

- Ubuntu 20.04.5 LTS[25]
- Linux Kernel 5.15.0-53-generic
- GNU bash 5.0.17(1)-release (x86_64-pc-linux-gnu)

[22] https://www.amd.com/en/product/7926

[23] https://www.nvidia.com/en-us/geforce/graphics-cards/30-series/rtx-3090-3090ti/

[24] https://documents.westerndigital.com/content/dam/doc-library/en_us/assets/public/western-digital/product/internal-drives/wd-blue-nvme-ssd/product-brief-wd-blue-sn550-nvme-ssd.pdf

[25] https://releases.ubuntu.com/focal/

- Python 3.8.10[26]
- R version 4.2.2 Patched (2022-11-10 r83330)[27]

[26] https://www.python.org/downloads/release/python-3810/

[27] https://cran.r-project.org/bin/linux/ubuntu/

Investigating the Importance of Demographic Features for EDM-Predictions

Lea Cohausz
University of Mannheim
lea.cohausz@uni-mannheim.de [*]

Andrej Tschalzev
University of Mannheim
andrej.tschalzev@uni-mannheim.de [*]

Christian Bartelt
University of Mannheim
christian.bartelt@uni-mannheim.de

Heiner Stuckenschmidt
University of Mannheim
heiner.stuckenschmidt@uni-mannheim.de

ABSTRACT

Demographic features are commonly used in Educational Data Mining (EDM) research to predict at-risk students. Yet, the practice of using demographic features has to be considered extremely problematic due to the data's sensitive nature, but also because (historic and representation) biases likely exist in the training data, which leads to strong fairness concerns. At the same time and despite the frequent use, the value of demographic features for prediction accuracy remains unclear. In this paper, we systematically investigate the importance of demographic features for at-risk prediction using several publicly available datasets from different countries. We find strong evidence that including demographic features does not lead to better-performing models as long as some study-related features exist, such as performance or activity data. Additionally, we show that models, nonetheless, place importance on these features when they are included in the data – although this is not necessary for accuracy. These findings, together with our discussion, strongly suggest that at-risk prediction should not include demographic features. Our code is available at: https://anonymous.4open.science/r/edm-F7D1.

Keywords

at-risk prediction, demographic features, fairness, bias, categorical features

1. INTRODUCTION

The use of demographic features for training models to predict at-risk students, e.g., students in danger of dropping out or failing a course or study program, is very common [2, 21]. Demographic features "refer to particular characteristics of a population [. . .], such as age, race, gender, ethnicity, religion, income, education, [. . .]" [25]. These features are typically categorical and sometimes also of high cardinality. Other features usually used in the context of at-risk prediction are previous performance features (e.g., previous results, current GPA, ...) as well as study engagement/activity data (e.g., log data, count of raised hands) [31]. Alturki et al. [2] evaluated the features most used across EDM studies predicting student success from 2007-2018. Among the ten most used features are six demographic features (gender, age, income, nationality, marital status, employment status) – the most common of which is gender. In a way, it is not surprising that these features are so regularly used. Most educational institutions require the students to enter demographic information about themselves, and this data is typically more accessible to researchers than, e.g., log data. However, demographic features also make datasets very problematic regarding receiving access and sharing the data [9]. Demographic data is sensitive data and can be used to identify people in the dataset. In order to be able to share the data, at least some type of pseudonymization has to be employed, e.g., k-anonymity [29]. This is often extremely difficult to achieve and weakens the usefulness of the features, e.g., through binning.

Apart from the problems with data access, demographic features are also problematic in some settings where we could employ the models. Suppose we, e.g., use a model to admit people to a course based on their prediction. In that case, it is very problematic if demographic variables impact the prediction as it could easily reproduce biases [13]. Due to these fairness concerns, the use of demographic features is heavily discussed in the literature on fairness in AI [17].

Hence, using demographic features in predictive models leads to a lot of problems. Still, if demographic features are relevant for EDM predictions, it might be tempting for researchers and practitioners to include them. Yet, their value for the prediction is unclear. Few papers explicitly evaluated feature importance, and even fewer considered the effect of demographic features in general. Those that have arrived at very different conclusions. While some stress the importance of demographic features [4, 7, 12], others state that they are not important [31, 32, 19], and others yet are on middle ground [15, 30, 35, 6]. So far, comparatively few papers compared the accuracy metrics of models with and without demographic information [31, 1, 14]. Furthermore,

[*] Both authors contributed equally to the paper

L. Cohausz, A. Tschalzev, C. Bartelt, and H. Stuckenschmidt. Investigating the importance of demographic features for edm-predictions. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 125–136, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115647

the nature of the relationship (linear, nonlinear) between demographic features and study success has not been evaluated enough. Suppose we find that demographic features are not important for model performance. In that case, it pays to leave them out – mostly for fairness and privacy reasons but also because wisely selecting features pays off regarding the amount of data instances we need to train our models, as more features require more instances [34]. In detail, the contributions of this paper are:

- We provide a theoretical discussion on the type of features typically used for at-risk prediction (Section 2.1), fairness concerns when using demographic features to predict academic achievement (Section 2.2), and causal mechanisms that could exist between demographic characteristics and academic achievement (Section 2.3).
- We summarize and discuss the findings of existing studies on the importance of demographic data in Section 3.
- We evaluate the importance of demographic features for predicting academic success using four EDM datasets in Section 6 and show that demographic characteristics are related to the target, but when study-related information is available, using them does not increase the predictive performance
- We find that models nonetheless place importance on demographic features when they are included such that practitioners cannot rely on technical solutions but have to carefully think about whether demographic features should be included at all in Section 7.

2. THEORETICAL CONSIDERATIONS

2.1 Types of Features

As already mentioned in the introduction, we can have different types of features in the datasets. In accordance with Tomasevic et al. [31] we argue that there are three major types of EDM features: demographic features, performance features, and activity/engagement features.

Demographic Features. Demographic features are traditionally considered to be features that refer to characteristics of a population. Typically used demographic features are gender, age, ethnicity, nationality, or features indicating socioeconomic status, such as e.g. parental occupations or household income. Furthermore, we define all features as demographic features that strongly point toward certain demographic characteristics. For example, we consider the school a student went to or parental financial support as demographic information.

Performance Features. Any study-related performance measures, e.g., grades, information on passes or fails, or percentages on assignments, are considered performance features. In other words, any information that hints on how well a student did in the past belongs to this type.

Activity Features. Activity features are features that are study-related and show how active a student is. Typical features of this type are participation during class, hours spent on online-learning platforms, participation in online forums, etc.

Most features in EDM datasets belong to one of these categories with the implicit assumption that they all matter regarding at-risk prediction. Other features not belonging to either of these categories would, e.g., be the study program or the semester a student is in or in which the course takes place. As our focus is on investigating whether using demographic features is advantageous when we also have some study-related features, we do not differentiate between activity and performance data. For the remainder of the paper, we define study-related features as all features related to a student's study activity and previous performance.

2.2 Fairness Considerations

Before we start investigating the potential usefulness of demographic features in detail, we want to briefly highlight why fairness concerns are so prevalent when it comes to demographic features and why it is so important to investigate their potential impact.

Most datasets used in EDM research consist of historical data. Historical data may already include biases [17]. If e.g., a teacher unconsciously or consciously favors students of a certain gender or ethnicity, students belonging to this demographic category will have better grades. A machine learning model will learn this pattern and, as a result, is more likely to predict that students who belong to different genders or ethnicities are at risk. If e.g., the prediction is used to admit students to a course or a degree, then it is very obvious that unfairness results from bias. Another probable problem in EDM research arises when some populations are underrepresented in the training data [17]. If, e.g., only one person in the data has children, and this person happens to perform badly, a machine learning model might simply learn that having children is a good predictor of bad performance. If we then predict how well another student with children will do, the model will likely predict them to be at risk. Again, the fairness concerns are obvious. This problem of underrepresentation may particularly occur when demographic features are categorical and of high cardinality, as fewer samples are available per categorical value. In this case, it is likely that some groups are poorly represented, and therefore, overfitting occurs, which can lead to bias.

One often-used strategy to circumvent these fairness issues is to completely remove obvious demographic features (such as ethnicity and gender) from the training data. Nonetheless, it is sometimes possible to still infer demographic information from other features that do not appear to be demographic features directly [17]. For example, if the school name is included in the training data, this might reveal the gender of a person ("ABC School for Girls"). The best strategy to avoid unfairness is, therefore, to try to exclude any features that point towards demographic characteristics and are not directly study-related when at-risk prediction models are deployed.

2.3 Causal Mechanisms including Demographic Features

Although it has hardly been done in EDM, it is important to consider how demographic features might causally impact study success theoretically. Understanding these mechanisms will help us to reason when demographic features may matter for the prediction but also again highlights why us-

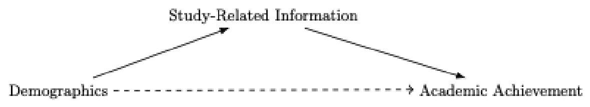


Figure 1: A graph to display the causal relationship between demographic aspects and the target.

ing demographic features for at-risk prediction is not ideal. Demographic features never directly impact study success but only through causal mechanisms. Drawing on social science literature, we classify those mechanisms into two types: capital-based and discrimination-based.

2.3.1 Capital-Based Mechanisms

Capital is typically divided into economic, social, and cultural capital [11]. Economic capital would, e.g., be money. If a student has little monetary means (low socioeconomic background), they might be forced to work a lot and live far away from campus. Working much and having to commute both means the student has less time to study, leading to less activity and poorer results. Social capital would, e.g., be to know whom to turn to if a student struggles or to have a social support network. Students from a low socioeconomic background or a foreign country might not have access to this knowledge and those connections. Similarly, people with such demographics might not know certain cultural rules (cultural capital) in academia which might also lead to disadvantages [23]. Capital-based mechanisms are diverse and probably exist in different settings, such as online and offline.

2.3.2 Discrimination-Based Mechanisms

Demographic features may also impact study success through discrimination, e.g., an instructor might consciously or unconsciously discriminate against students with certain characteristics. This could either directly impact a student's academic achievement or indirectly as the student perceives the discrimination and reacts by spending less time and effort on the course [20]. The effect of discrimination-based mechanisms should vary from setting to setting. For example, discrimination could be less likely in online settings as teachers do not receive visual cues regarding students' demographic characteristics.

2.3.3 Mediation Effects

Overall, capital-based mechanisms probably exist universally. However, once someone is in higher education, demographic characteristics will have already impacted previous performance (in school and then in previous university courses). This might mean that as long as we have information on previous performances, demographic data has no additional effect. Demographic characteristics might also impact a student's activity. Not having time naturally leads to less study engagement. Furthermore, e.g., people from a lower socioeconomic background might also be hesitant to participate in class. So, again, having activity information may – at least partly – make demographic data redundant. To a degree, these considerations might also be true for discrimination-based mechanisms; here, however, the effect of demographic characteristics should vary between courses and also between different settings, e.g., between different universities and online and offline learning. Returning to our discussion on fairness, it is also obvious that ML models in action should not predict based on previous discrimination against certain populations.

In summary, demographic features are causally related to study success and may, therefore, be important for predictions. However, their impact is likely already captured by previous performance and, potentially, to a degree, previous activity data, such that the performance gain in using them for predictions is small too not existent. In other words, other study-related features mediate the effect of demographic characteristics. This mechanism can also be seen in Figure 1.

3. EXISTING EVIDENCE

As already mentioned, existing research is divided on whether demographic features are important for predictions or not. In this section, we will look at contributions highlighting the importance, lack thereof, or some middle ground between these stands.

3.1 Demographic Features Are Important

Batool et al. [4] used the widely used Open University Learning Analytics Dataset (OULAD) and two similarly structured datasets and used only the demographic features in the datasets to predict who will fail the courses. They report high F1-scores using Random Forests but do not compare against baselines to validate the meaningfulness of their results. Daud et al. [7] predict whether a student will finish their degree based on socio-demographic features using a dataset from several universities in Pakistan. They considered many features not typically available and potentially extremely problematic such as e.g., family expenditures. Daud et al. report high F1-scores, with their best method generally being the Support Vector Machine followed by Naive Bayes, but do not compare this against predictions using previous performance data. Hoffait et al. [12] predict which students are at-risk at the time of registration for their degree using a dataset from Belgium. Due to their setting, they only have some previous performance data from school and no activity information, but most of their data is demographic. Yet, they achieve relatively high F1-scores. Their Random Forest model slightly outperforms other models, such as Logistic regression or a Neural Network.

3.2 Demographic Features Are Not Important

Tomasevic et al. [31] also used the OULAD to predict performance and compared several Machine Learning models with different sets (demographic, performance, activity) of features against each other in a very thorough study. Usually, the prediction accuracy did not vary much when using or not using demographic features as long as the other sets of study-related features were used, leading them to conclude that these features were not important, although using demographic features usually slightly improved the model. At least for this dataset, this is very strong evidence that demographic features do not significantly add to the prediction accuracy. It should be noted that they apparently did not use all demographic features available. Their best-performing model was the Neural Network. Al-Zawqari and Vandersteen [1] use a subset of the OULAD dataset to distinguish between high-performing and failing students. They compared F1-scores using and not using demographic data along with activity data and found that using demographic data did not improve results much. It should be noted that it is unclear how they selected and handled their data. Random Forests and Neural Networks performed almost equally well. Jha et al. [14] used the same dataset to predict failure using a variety of methods and different feature subsets. In accordance with the other papers, they found that activity data was the most predictive feature set. When they used activity data, it did not matter what other features were included regarding the model's performance. Trstenjak and Donko [32] used data from the Information System of Higher Education Institutions databases, predicted success using Support Vector Machines and Naive Bayes, and ranked feature importance using several metrics such as information gain and gain ratio. They showed that most (but not all) demographic features had very little impact and experimented with leaving some (the least important ones) of them out, which even led to slightly increased accuracy. Support Vector Machines outperformed Naive Bayes. Miguéis et al. [19] predicted the overall study success of students of a technical university and then looked at the Gini-index of features. They found that performance data was more important than demographic data, with AdaBoost being their most accurate model.

3.3 Demographic Features Are Somewhat Important

Khasanah et al. [15] predicted overall study success with data from Indonesia using Decision Trees and Bayesian Networks, with Bayesian Networks being more accurate. They used Information Gain to evaluate demographic feature importance and found that some were important, but others were not. It should be noted that the data they had available on previous performance and activity was rather limited. Sweeney et al. [30] looked at the feature importance of one large dataset as they tried to predict study success using a Factorization Machine for the courses a student enrolled in the next term. They found that demographic data were more important in the beginning when little past performance data was available than later on. They had relatively few demographic features in their dataset, however. Zhao et al. [35] use admissions data to predict who will perform well in a specific Master's program based on admission data. Due to the nature of their setting – that they try to learn who should be admitted to the program – their performance data is restricted to data on high school and Bachelor results, and they have no activity data. Though they make no difference between demographic and non-demographic features, their most important predictors show that some demographic features (gender, nationality) tend to be important while others are not. Random Forest or ensemble methods tend to be the best-performing models. Cortez and Silva [6] predicted grades of Portuguese middle school students in math and Portuguese. They found that the relative importance of previous performance scores was higher, but socio-demographic features still mattered. They provide a detailed list of their preprocessing, typically including binning or ordinal recoding. Random Forests tended to perform best.

3.4 Overall Evidence

Overall, for the case of OULAD, despite Batool et al.'s results [4], the evidence appears to be pretty clear that accuracy does not increase when using demographic data along with performance or activity data [31, 14]. In general, studies that included study-related features typically found demographic features to be less important. However, in other settings where fewer performance data is available, results suggest that demographic data does play a role. Those that explicitly investigated feature importance typically reported that it is somewhat important. Furthermore, note that only very few studies explicitly reported on feature engineering of demographic characteristics. Yet, feature engineering is often non-trivial for demographic data as it often consists of (high-cardinality) categorical data.

4. RESEARCH OBJECTIVE AND QUESTIONS

Both our review of existing evidence and our theoretical considerations lead us to the hypothesis that using demographic features will not increase model performance as long as we have study-related features from previous performance or activity but that they will have predictive power if we do not have study-related features.

To test our hypothesis, we formulate the following main research questions:

- **RQ1**: Are demographic characteristics useful in explaining at least some of the differences in student performance; in other words, are models using only demographic features better than guessing?
- **RQ2**: Are demographic characteristics still useful if study-related information is available; in other words, do models trained on study-related and demographic features perform better than models trained only on study-related features?
- **RQ3**: Which features should ultimately be used in EDM predictions; in other words, models trained on which feature subsets outperform models trained on other subsets?
- **RQ4**: If **RQ2** is answered with no, do models trained on the whole data learn that demographic information is irrelevant; in other words, do models trained on the whole data place close to zero importance on the demographic features?

Furthermore, we are interested in the following research questions:

- **RQ5**: How complex is the relationship between predictive features and student performance; in other words, how large are the differences in performance between linear and nonlinear models?
- **RQ6**: How relevant is the treatment of categorical features; in other words, do different encoding methods affect performance?

5. EXPERIMENTAL DESIGN

In this section, we describe our experimental setup to evaluate the formulated research questions. We proceed by first describing the used datasets and model classes used for prediction. Afterward, the hyperparameter tuning procedure, methods to treat categorical data, and the evaluation setup are described.

5.1 Datasets

We use four publicly available EDM datasets. Two datasets are from online learning systems and two from in-class education, of which one is from secondary education in high schools and one from tertiary university education. In this subsection, we briefly describe the used datasets and the corresponding preprocessing. Furthermore, we describe the assignment of features to the feature types (demographic, performance-related, activity-related, and others) discussed in Subsection 2.1. We will use the resulting feature subsets in Section 6 to train models for answering the research questions. An overview of the datasets can be seen in Table 1.

5.1.1 Dataset of Academic Performance Evolution for Engineering Students

The dataset of academic performance evolution for engineering students [8] consists of the academic, social, and economic information of $12,411$ Columbian engineering students. Student performance was assessed at two points in time: in the final year of high school and in the final year of pursuing a professional career in Engineering. We refer to this dataset as *Engineering*. The first assessment evaluates five generic academic competencies: mathematics, critical reading, citizen competencies, biology, and English. The second assessment evaluates critical reading, quantitative reasoning, citizen competencies, written communication, English, and the formulation of engineering projects. As the target for predictions, we use the global score of the second performance assessment and treat the task as a regression task. The five dimensions of the first assessment are used as performance information. There is no information about student activity in the dataset. Demographic features include gender, parental education and occupation, geographic information, school information, and whether different items, such as a car or computer, were available in the family. Other available information is the university and the academic program a student attends. The identifier features, as well as all dimensions and variants of the performance assessment besides the global score, are excluded. Further dataset-specific preprocessing is not necessary. Thirteen categorical features are in the dataset, of which two are of very high cardinality. There are students from 3,735 schools and 134 universities.

5.1.2 Dataset of Portuguese Secondary School Student Performance

The dataset [6] consists of students from secondary education in two Portuguese schools and can be used to predict student achievement in math and Portuguese language courses. We refer to this dataset as *PortSecStud* The target is the final course grade, which is measured on a discrete scale between 0 and 20. Some authors categorize the grade into pass and fail for binary classification or into five levels for classification. However, we consider it a regression problem, as it better represents the nature of the problem. As performance information, the first and second-period grade is available, as well as the number of past class failures. Activity information consists of the weekly study time, absences, and whether the student participated in extracurricular activities. The demographic information includes gender, age, and address, as well as school and family-related information. Furthermore, we considered travel time from home to school, educational support from family, extra paid classes within the course subject, and having internet access as demographic features since they are highly influenced by socioeconomic factors. Other features are lifestyle-related ones such as alcohol consumption, health status, or whether the student is engaged in a romantic relationship. The datasets for the math and Portuguese courses are combined and a feature indicating the course is added. Further dataset-specific preprocessing is not necessary.

5.1.3 xAPI-Edu-Data

The Students' Academic Performance Dataset (xAPI-Edu-Data) [3] consists of 480 students, where most are from Kuwait (179) and Jordan (172). The target is students' performance in %, which is only available in groups: 0-69, 70-89, and 90-100. Hence, we treat the task as a multi-class classification problem. There is no information about previous student performance in the dataset. Student activity is measured according to four behavioral aspects during interactions with the e-learning system: participation in discussion groups, visiting resources, raising a hand in class, and viewing announcements. In addition, absence days are available. Demographical features are nationality, gender, place of birth, and the parent responsible for the student. Other information includes the academic background (e.g., course, semester, grade level), and the parents' participation (answering a survey, school satisfaction). No dataset-specific preprocessing is required. The categorical features with the most expressions are nationality, with 14 possible nationalities, and field of study, with 12 possible subjects.

5.1.4 OULAD

The OULAD dataset is a large dataset with diverse opportunities for educational data mining [16]. It is a relational database of five tables with information on students, assessments, courses, registrations, online learning materials, and students' interactions with those. We focus on the same prediction task with the same dataset, features, and preprocessing as Jha et al. (2019) [14]. For predictions, we consider all students who did not drop out before the course ended to predict whether they failed or passed. As information about the previous performance, we use the average scores achieved in previous assignments. Jha et al. (2019) [14] conducted analyses on different data subsets as well; however, they counted the so-far achieved credits and the number of previous attempts as demographic features. This does not match our definition of demographic features. Hence we define those features as performance-related. Student activity is obtained as two types of interaction with 20 different content types resulting in 40 features. The types of interaction are the sum of the clicks and the number of visits for each type of content. Examples of content types are homepage, subpage, quiz, wiki, and other platform-related types. As demographic features, we use gender, region,

Table 1: Description of the used datasets.

	Engineering [8]	PortSecStud [6]	xAPI-Edu [3]	OULAD [16]
No. of samples	12411	1044	480	22437
No. of features	33	34	17	51
Performance features	5	3	0	4
Demographic features	25	17	4	6
Activity features	0	6	5	40
Other features	2	7	7	0
Categorical features	13	4	7	4
Total cardinality	3980	17	59	31
% NA	0.0	0	0.0	0.48
Target $\mathbf{y} \in$	[1..166]	[1..19]	[1..3]	{1,2}

imd_band, age_band, and disability. There are no other features in the dataset. The performance and activity features are extracted from the database as described by Jha et al. (2019) [14]. Similarly, the id_student, code_module, module_presentation, and exam_score features were excluded as well as all students who had withdrawn before the course ended. Some mean assessment scores and imd_band categories are missing. As the information on how missing values are treated is not given in [14], we impute the mean value for the mean assessment scores and define a new category for missing imd_band values.

5.2 Models

We include two model classes, namely generalized linear models (GLMs) and XGBoost, in our evaluation. For regression tasks, we use Lasso regression for the regularization of the models to prevent overfitting. For classification tasks, we use logistic regression with the L2-penalty. In the case of multi-class classification, multinomial loss is used. GLMs have the benefit of being highly interpretable and, thus, are ideally suited for (educational) data mining. However, they make the strong assumption that the relationship of the target to the features is linear. In contrast, XGBoost is a highly flexible model capable of learning more complex relationships. For the OULAD dataset, XGBoost has been shown to outperform competitive approaches by Jha et al. (2019) [14]. Furthermore, for tabular datasets, XGBoost has shown superior performance compared to other methods like neural networks far beyond the field of educational data mining [26, 10]. Thus, it can be considered the state-of-the-art model for maximizing performance on a variety of datasets such that we do not include further models. By comparing the predictive performance of GLMs and XGBoost, we are able to answer research question **RQ5**. In addition, baseline models for each dataset are included, which predict the target mean of the training data for regression tasks and the mode for classification tasks. By comparing models trained solely on demographic data to these baselines, we are able to answer research question **RQ1**.

5.3 Hyperparameter Optimization

We implement a hyperparameter optimization (HPO) pipeline with 5-fold cross-validation (5CV) for XGB and GLMs. For parameter tuning, we use Bayesian optimization implemented in the hyperopt library [5]. To select the best parameters, the training data is split into five folds again. In each HPO step, a model with the current hyperparameters is trained on each fold. The objective function of each step is the average performance on the held-out datasets of each fold. Our modeling pipeline is depicted in Figure 2. Performance is measured as the mean squared error (MSE) for regression tasks and log-loss for classification tasks. For the GLMs, we only tune the regularization strength parameter α. The search space for Lasso regression is defined as $\alpha \in [10^{-10}, 0.5]$. The search space for Logistic regression is defined as $\alpha \in [10^{-10}, 1.0]$. We run 50 iterations of Bayesian optimization for each model. For hyperparameter optimization of XGBoost, we implement an algorithm to iteratively tune different subsets of XGBoost hyperparameters using Bayesian optimization in four steps. (1) Tune the number of estimators $\in [50..500]$ and the learning rate $\in [0.001, 0.5]$. (2) Tune the maximum tree depth $\in [1..18]$ and minimum child weight $\in [0..10]$. (3) Tune both the number of columns and samples used in each tree $\in [0.5, 1]$. (4) Tune the regularization parameters $\alpha \in [0..10]$, $\lambda \in [1, 4]$ and $\gamma \in [10^{-8}, 9]$. In each step, 50 iterations of Bayesian optimization are performed. To speed up the computations and terminate the training for optimization iterations with poor parameter choices more quickly, we use early stopping on the validation data if there is no improvement after ten training iterations. Overfitting on the validation data is mitigated through the 5CV procedure as a configuration needs to perform well on all five validation sets.

5.4 Methods for Categorical Data Treatment

All of the used datasets include categorical data. As the treatment of categorical data can affect predictive performance in data mining tasks [22], we want to evaluate whether our models are affected by different encoding methods. Hence, to answer research question **RQ6**, we evaluate if and how much different encodings of categorical features impact the prediction. Each categorical feature with three or more unique values is considered. Ordinal features are treated as categorical as well. One-Hot-Encoding (OHE) is included as it is the most frequently used method to handle categorical data. Categories in the test data which did not appear in the train data are ignored such that the encoding vector consists solely of 0s. As sometimes categorical features can be of high cardinality, OHE can suffer from overparameterization and unnecessary sparsity, leading to increased training times and memory requirements. Therefore, we include ordinal encoding as it can be a simple and more compact encoding and is frequently used for XGBoost. However, for linear models, ordinal encoding is not appropriate as there is no natural order between the categories. Unknown values are encoded in a new category. A generally applicable

Table 2: Description of the allocations of features to subsets.

	Demographic Features	Study-Related Features	Other Features
Engineering [8]	gender, parental, geographic, and school information, item availability in family	first assessment on five dimensions (MAT, CR, CC, BIO, ENG)	university, academic program
PortSecStud [6]	gender, age, address, family and school related information, paid classes, internet access	first and second period grade, past failures, absences, study time, extracurricular activities	lifestyle related features, e.g. alcohol consumption, romantic relationships, amount of free time
xAPI-Edu [3]	gender, nationality, place of birth, parent responsible	interaction with the e-learning system, absences	general academic information (e.g. semester), parental participation
OULAD [16]	Gender, region, imd_band, age_band, disability, highest_education	num_of_prev_attempts, avg_cma, avg_tma, studied_credits, sum of clicks and count of visits for each of the 20 VLE activity types	-

method is target encoding and its variants [18]. In target encoding for regression, each categorical value is encoded as the target mean of the training samples belonging to this value. For classification, the posterior probability of the target given the categorical value is used. As this approach is sensitive to overfitting, the encoding is further blended with the global mean value for regression and the prior target probability for classification. In the case of multi-class classification, we use a one vs. rest approach to obtain an encoding for each class. For unknown categories in the test data, the global mean or prior probability is used. In addition, the Catboost encoder is included as it was specially designed for improving categorical data handling in gradient boosting [24]. The method is similar to target encoding but considers the frequency counts of expressions of a categorical feature in a more principled way. For high-cardinality features, regularized target encoding was shown to be the superior method for a variety of datasets in a large benchmark study [22]. Therefore, we also include 5CV-GLMM, the best-performing method from that study, in our evaluation. The method first fits a simple generalized linear mixed model (GLMM) for each categorical feature and uses the estimated random effects coefficients of the model as encodings. To prevent overfitting, this procedure is combined with 5-fold cross-validation (5CV). The train data is separated into five parts, and five GLMMs are fitted to 80% of the data, and the estimated random effects of the model are used as encodings for the remaining data. The test data is encoded using a model trained on the whole train data. We implement 5CV-GLMM encoding using the gpboost library [27, 28] as it provides a very efficient implementation of GLMMs.

5.5 Predictive Performance Evaluation

For regression tasks, the target is normalized to zero mean and unit variance for training the models, and the predictions are denormalized afterward to interpret the performance on the original scale. All continuous features were normalized to zero mean and unit variance as well to be able to interpret GLM coefficients as feature importance. For each configuration, we use 5-fold cross-validation (5CV) for evaluation. As an evaluation metric, we use root mean squared error (RMSE) for regression (lower is better) and F1-score with macro averaging for classification (higher is better).

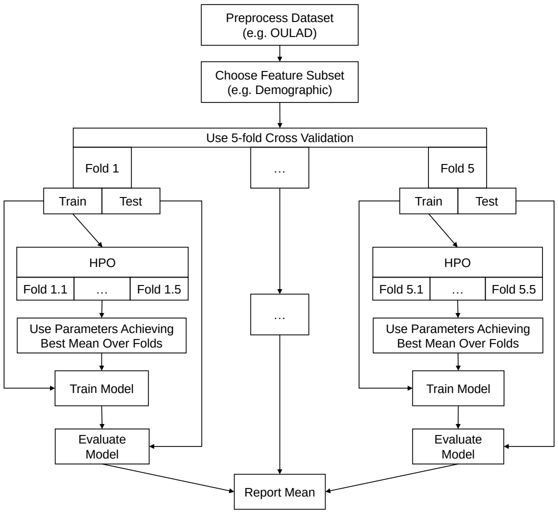


Figure 2: Data pipeline for model development and evaluation.

5.6 Methods for Determining the Impact of Demographic Features in Models

As we investigate the importance of demographic data in EDM predictions, performance is not the only relevant metric. It is equally important to analyze the extent to which the models use demographic data. Hence, to answer our research question **RQ4**, we analyze the feature importance of trained models with a focus on demographic data. We analyze the learned coefficients of the linear models as well as the feature importance of the XGBoost models. As we normalized the data, the coefficients of linear models can directly be interpreted as feature importance scores. For the linear models, we first normalize the absolute coefficient values to sum to one. Afterward, we sum the normalized coefficients for the demographic features to obtain an assessment of the extent to which the models use demographic data for predictions. For XGBoost the feature importances reflect how often certain features are used and how useful they are for the prediction in a single decision tree. Precisely, the importance of a single tree is calculated as the amount that each split improves the performance measure, weighted by the number of samples the node is responsible for. Afterward, the feature importances are averaged across all of the decision trees and normalized to sum to one. In addition, we analyze the extent to which the utilization of demographic data affects the actual predictions on linear models; this is considered important information in the fairness literature [17]. Given a dataset with n samples and d features in a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ and a target $\mathbf{y} \in \mathbb{R}^n$, we apply the following procedure:

1. Train a linear model to predict the target
2. Obtain predictions as $\hat{\mathbf{y}} = \sigma(\mathbf{X}\boldsymbol{\beta})$, where $\boldsymbol{\beta}$ is the coefficient vector of the linear model and σ is the inverse link function depending on the target, e.g., linear for continuous and sigmoid for binary targets
3. Remove the k demographic features from $\mathbf{X}$ and the respective coefficients $\boldsymbol{\beta}$ and obtain predictions $\tilde{\mathbf{y}} = \sigma(\mathbf{X}_{:,d-k}\boldsymbol{\beta}_{d-k})$
4. Compute score for the impact of demographic features as
   (a) $\frac{1}{n}\sum_i^n \hat{y}_i - \tilde{y}_i$ for regression
   (b) $\frac{1}{n}\sum_i^n \{1 \text{ if } \hat{y}_i \neq \tilde{y}_i, 0 \text{ otherwise } \}$ for classification

For regression, this corresponds to evaluating the mean absolute difference of predictions with and without demographic features. For classification, this corresponds to evaluating the percentage of samples for which not using the demographic features changes the class assignment.

6. RESULTS

In this section, we report and discuss our results to evaluate the stated research questions. We start with analyzing the effect of the categorical data treatment method. Afterward, we proceed with a comparison of models trained on different data subsets with a focus on demographic data. Finally, we evaluate the feature importance of the final models to assess whether demographic data is used.

6.1 Impact of Categorical Data Treatment Method

Table 3 shows the results for the different datasets and categorical data treatment methods. In general, it appears that the treatment method does not matter. According to a t-test over the folds, for three datasets, ignoring the categorical features works just as well as using the categorical data, regardless of the encoding method. For the Engineering dataset, the target scale is rather large, such that performance differences on the digits after the decimal point do not matter. Hence, for this dataset, too, we can consider the encoding method irrelevant. As most of the categorical features are demographic features, this indicates a low importance of demographic characteristics. Finally, our conclusion to **RQ6** is that the treatment of categorical features does not affect performance. Hence, we use 5CV-GLMM encoding for GLMs and ordinal encoding for XGBoost in the following experiments.

6.2 Performance Comparison of Different Feature Subsets

Table 4 shows the results for different data subsets as defined in section 5.1.

6.2.1 Predictive Capability of Demographic Features

For OULAD, no difference can be seen between the baseline and using solely demographic features for prediction. Hence, predicting that every student passes the course works equally well as training a model solely using demographic features. For the PortSecStud dataset, the improvement over the baseline is small, such that the usefulness of the demographic features can be considered small also for this dataset. For the Engineering dataset, there is a considerable improvement over the baseline, and for the xAPI-Edu-Data, the improvement over the baseline is the largest. Hence, for these two datasets, it can be said that there is an impact of the demographic features on the performance. Considering **RQ1**, we conclude that demographic characteristics can be used to explain differences in student achievement. However, this does not hold in every setting and for every type of demographic characteristic.

6.2.2 Mediation Capability of Study-Related Features

For all datasets, using only study-related features achieves far better performance than using only demographic features. Using only study-related features achieves approximately the same performance as additionally considering demographic features in almost every setting. Only for XGBoost on the xAPI-Edu-Data, there is a noteworthy mean difference; however, given the large standard deviations, it cannot be said that it is meaningful in practice. These results confirm the hypothesis that study-related information mediates the effect of demographic characteristics on student achievement. As soon as meaningful information about the student's activity and/or previous performance is available, the demographic features are not required anymore for accurate predictions. Hence, considering **RQ2**, demographic characteristics are generally not useful anymore if study-related information is available.

6.2.3 Feature Subsets Achieving the Best Performance

It can be seen that for all datasets, using the whole data is always among the best subsets, as can be expected. Furthermore, according to the t-test for the OULAD dataset, using only study-related information performs equally well as using the whole available information for both GLMs and XGB. The same holds for XGB on the PortSecStud dataset and GLMs on the xAPI-Edu-Data. Given the range of the target values of the Engineering and the PortSecStud datasets, the performance differences are not meaningful in practice, despite the significant differences in the t-test. We conclude that using only study-related features suffices as well for this dataset. For XGBoost on the xAPI-Edu-Data, using all features performs significantly better than solely using study-related features. However, this is rather due to the other features included in the dataset than due to the demographic features, as the performance increase in using demographic data in addition to study-related data is insignificant. Hence, also for this dataset, it would be suitable not to use demographic features without significant loss of performance. In summary, models using all except

Table 3: Means and standard deviations of 5CV Performance results on different data subsets. Mean squared error is reported for Engineering and PortSecStud and F1-score for xAPI-Edu and OULAD. Results per row for methods that are not significantly different from the best method in a paired t-test (alpha=0.05) are highlighted in bold.

Dataset		Baseline	Ignore	OHE	Target	Ordinal	Catboost	5CV-GLMM
Engineering	GLM	23.11 (0.26)	14.36 (0.26)	**14.11 (0.29)**	14.55 (0.23)	14.36 (0.25)	14.22 (0.29)	**14.13 (0.29)**
	XGB	23.11 (0.26)	14.28 (0.25)	14.11 (0.26)	14.38 (0.23)	14.15 (0.25)	14.17 (0.28)	**14.04 (0.28)**
PortSecStud	GLM	3.86 (0.17)	**1.55 (0.1)**	**1.55 (0.1)**	**1.55 (0.1)**	**1.55 (0.1)**	**1.55 (0.1)**	**1.55 (0.1)**
	XGB	3.86 (0.17)	**1.51 (0.12)**	1.51 (0.08)	**1.46 (0.06)**	**1.52 (0.06)**	**1.57 (0.08)**	**1.5 (0.06)**
xAPI-Edu	GLM	0.2 (0.02)	**0.75 (0.06)**	**0.76 (0.03)**	**0.75 (0.06)**	**0.73 (0.05)**	**0.73 (0.09)**	**0.75 (0.06)**
	XGB	0.2 (0.02)	**0.76 (0.05)**	**0.78 (0.04)**	**0.78 (0.06)**	**0.78 (0.08)**	0.72 (0.07)	**0.76 (0.02)**
OULAD	GLM	0.81 (0.0)	**0.87 (0.01)**	**0.87 (0.01)**	**0.87 (0.01)**	**0.87 (0.01)**	**0.87 (0.01)**	**0.87 (0.01)**
	XGB	0.81 (0.0)	**0.91 (0.0)**	**0.91 (0.0)**	**0.91 (0.0)**	**0.91 (0.0)**	**0.91 (0.0)**	**0.91 (0.0)**

Table 4: Means and standard deviations of 5CV Performance results on different data subsets. Mean squared error is reported for Engineering and PortSecStud and F1-score for xAPI-Edu and OULAD. Results per column for methods that are not significantly different from the best method in a paired t-test (alpha=0.05) are highlighted in bold.

Dataset		Baseline	Demo only	Study only	Demo + Study	All
Engineering	GLM	23.11 (0.26)	20.53 (0.3)	14.47 (0.22)	14.35 (0.25)	**14.14 (0.29)**
	XGB	23.11 (0.26)	20.43 (0.34)	14.39 (0.2)	14.28 (0.23)	**14.05 (0.29)**
PortSecStud	GLM	3.86 (0.17)	3.76 (0.11)	1.57 (0.1)	1.58 (0.1)	**1.56 (0.1)**
	XGB	3.86 (0.17)	3.83 (0.15)	**1.49 (0.09)**	**1.54 (0.14)**	**1.54 (0.05)**
xAPI-Edu	GLM	0.2 (0.02)	0.39 (0.04)	**0.74 (0.03)**	**0.74 (0.05)**	**0.74 (0.06)**
	XGB	0.2 (0.02)	0.54 (0.03)	0.74 (0.03)	**0.75 (0.05)**	**0.78 (0.05)**
OULAD	GLM	0.81 (0.0)	0.81 (0.0)	**0.87 (0.01)**	**0.87 (0.01)**	**0.87 (0.01)**
	XGB	0.81 (0.0)	0.81 (0.0)	**0.91 (0.0)**	**0.91 (0.0)**	**0.91 (0.0)**

the demographic features do not perform significantly worse than models additionally considering demographic features. Given the sensitive nature of demographic features, we conclude **RQ3** with the recommendation to use only study-related and other than demographic features for predicting student success. If sufficient study-related information is not available but predictive performance matters, demographic features may still be helpful.

6.2.4 Comparison between GLMs and XGBoost

For the regression datasets, the difference between GLMs and XGBoost is small for all models, such that we would prefer GLMs as the simpler solution. For the xAPI-Edu-Data [3], XGBoost is superior for models trained on the whole data on average. However, this has to be viewed with care as the standard deviation between folds is large. Furthermore, GLMs perform equally well as XGBoost when using only the study-related data. Hence, using GLMs solely on performance and activity data could be an alternative for this dataset. For the OULAD dataset, there is a clear performance benefit in using nonlinear methods. As the dataset is large, the results are more robust, with a small standard deviation between folds. Hence, we can say that for this dataset, using XGBoost solely on activity and performance data would be the preferred solution. Considering **RQ5**, there is evidence that for some educational data mining datasets, using linear models for at-risk prediction suffices. However, when larger datasets with thousands of students are available, nonlinear methods can perform better. These datasets can especially be collected in online settings similar to the OULAD datasets. However, for small in-class datasets, linear models should be the first choice.

6.3 Feature Importance of Demographic Data

The previous subsections have provided clear evidence that demographic features are not necessary for at-risk predictions when sufficient information about students' study activities or previous performance is available. However, our theoretical considerations indicate that demographic features might correlate with other study-related features. Thus, it is possible that models use these demographic features when they are included in the training data. To further inspect whether the tuned models learn that demographic features are not necessary for high predictive performance, we analyze the learned coefficients of the linear models as well as the feature importances of the XGBoost models as described in Subsection 5.6. Surprisingly, Table 5 shows that despite the fact that an equally good model could have been learned for all models without demographic features, those are still used for all models and datasets. Even for the PortSecStud dataset and the OULAD dataset, where we previously found that demographic features do not help at all compared to the naive baseline, the features are still used. For the XGBoost model trained on study-related and demographic data of the PortSecStud dataset, the demographic information even accounts for 26% of the feature importance despite not being necessary to achieve the performance. Furthermore, Table 6 shows that in every case, the utilization of demographic data directly affects the predictions of the models. For regression, the effect is not large considering the scales of the targets. Nevertheless, it might lead to biases for some students. For classification, the impact of demographic features on actual predictions is large. In general, if practitioners would be to use these models and look for an explanation for predictions, demographic features would be included, although this is not necessary.

Table 5: Means and standard deviations of relative feature importances of demographic data compared to the rest of the data in the model on different data subsets over all folds.

Dataset		Demo only	Demo + Study	All
Engineering	GLM	1.0 (0.0)	0.25 (0.03)	0.23 (0.02)
	XGB	1.0 (0.0)	0.23 (0.06)	0.19 (0.09)
PortSecStud	GLM	1.0 (0.0)	0.08 (0.04)	0.03 (0.02)
	XGB	1.0 (0.0)	0.26 (0.04)	0.16 (0.04)
xAPI-Edu	GLM	1.0 (0.0)	0.32 (0.14)	0.23 (0.1)
	XGB	1.0 (0.0)	0.26 (0.03)	0.21 (0.02)
OULAD	GLM	1.0 (0.0)	0.13 (0.0)	0.13 (0.0)
	XGB	1.0 (0.0)	0.08 (0.0)	0.08 (0.01)

Table 6: Means and standard deviations of the effect of demographic features on the predictions over all folds. For regression datasets, the mean absolute difference between predictions with and without demographic data is reported. For classification datasets, the percentage of predictions that change when excluding the demographic features from the model is reported.

Dataset	Effect of demographics
Engineering	0.97 (0.07)
PortSecStud	0.28 (0.06)
xAPI-Edu	0.11 (0.05)
OULAD	0.33 (0.01)

One might think that the unnecessary use of demographic features is related to our extensive hyperparameter optimization, which chose a hyperparameter configuration that considered all features, although it is not significantly better than another model with fewer features could be. However, we found that using the default configurations for sklearn linear models and XGBoost, either reproduces the same patterns or is not applicable due to bad performance. Hence, some kind of parameter selection or tuning is necessary. Generally, the models could be prevented from using all features by adjusting the regularization parameters accordingly. However, automatic parameter tuning does not guarantee finding this solution, as different parameterizations might achieve equal performance. In any case, our answer to **RQ4** is that just throwing in all features leads to models which use information from demographic features, although this is not necessary. That, combined with the previous results of this section, leads us to the general recommendation to consider completely leaving demographic features out for at-risk predictions whenever sufficient activity and/or previous performance information is available.

7. DISCUSSION

Our evaluation shows that using demographic features does not lead to better model performance as long as we include study-related features. Considering the fairness and privacy concerns, it is, thus, strongly advisable for both researchers and practitioners not to use these features for at-risk prediction.

7.1 The Importance of Demographic Features

Of course, this does not mean that we should never explore the impact of demographic features on academic achievement or that demographic features are not important. On the contrary, it is very important to investigate how demographic characteristics impact academic achievement so that we can intercept mechanisms that would lead to disadvantages of certain populations [21]. For example: If we notice that people from a lower socioeconomic background tend to a) live further away from campus and b) have to work a lot and that both of these influence the time they have for studying, which in turn influences their academic achievement, then we can come up with solutions for this on several levels. For example, the study management and academic staff might be able to come up with an adjusted study program and timetable. The university might provide cheap student housing close to the university; the state might provide funding for disadvantaged students.

Hence, we certainly do not want to discourage research on causal mechanisms of demographic characteristics on academic achievement. Rather, we want to highlight the importance of thinking about demographic features. Looking at the fairness literature, two aspects need to be highlighted. First, it might still be helpful to have demographic features available. This allows us to estimate a model's Demographic Parity Ratio or other common metrics in evaluating the fairness of a model's prediction [17]. Second, our theoretical model in 1 indicates that there might be proxy features that transport information about demographic features even if we do not include demographic features. It could, therefore, also be argued that demographic biases in these proxy features should be mitigated to receive a truly fair model [17]. Then, we would also need the demographic features to perform a form of bias mitigation.

7.2 Drawbacks of Feature Selection and Feature Attribution Methods

Most common feature selection or feature attribution techniques rely on the correlation between features and the target in one way or another. As we have discussed at length by now, demographic features are, in general, correlated with the target. This means that employing feature selection methods that rely on correlation will most likely lead to the inclusion of at least some of these features. Likewise, models are likely to place importance on demographic features as there are several equally good parameterizations leading to feature attribution methods recognizing these features as important. It also explains why some scholars who used such techniques (e.g., Information Gain or relative feature importance) reported that demographic features were at least partially important [15, 6, 35]. However, our analysis shows that it is not enough to simply employ such techniques to assess the usefulness of demographic features. While they are not necessary to achieve the best-performing model, they are still correlated with the target. Therefore, we recommend that researchers consciously think about whether and why they should include demographic features instead of using automatic (correlation-based) techniques for feature selection. Furthermore, in addition to feature attribution techniques, researchers should evaluate whether similar performance can be reached without certain features.

7.3 Implications for Practitioners

This is also one of the major implications of our paper for practitioners. Educational Data Mining researchers and practitioners should distinguish between models trained for deployment, where the goal is to achieve maximum performance, and models trained for gaining insights about the factors driving academic success. Depending on the application, the feature subset, especially whether to include demographic information, should be determined. Practitioners, when deploying models, should be very careful when it comes to including demographic features. They can, as discussed, not rely on technical solutions but should instead think critically about including these features. There may exist cases for which including demographic features is meaningful, but practitioners should be absolutely certain that including these features does not introduce biases producing unfairness. [33] state that sensitive features should be included for fairness reasons if the prediction accuracy itself is equal. However, Table 6 shows that using demographic features changed the prediction for some students; this is an indication that the models using demographic features are, indeed, unfair [17]. Therefore, leaving them in would probably not result in an equally fair model. Again, whether and how to use demographic features has to be carefully evaluated.

The other major recommendation for practitioners resulting from our paper concerns the kind of data useful for at-risk prediction. Our analysis clearly shows that past performance is extremely important, while demographic characteristics alone have little predictive power. In particular, features that mirror the requirements necessary to perform well in the target are highly relevant for the prediction. Therefore, practitioners should ideally use standardized tests that capture the kind of abilities relevant to the target. This would provide the best features for at-risk prediction.

7.4 Limitations of Our Study and Future Work

Despite our solid results, it is important to note certain limitations. We only used four datasets and two types of models to test our hypotheses. As these datasets are diverse (online, offline, different countries, different levels of education) as are the model types (linear, non-linear), we believe that our main findings are still very reliable. Nonetheless, future work should investigate whether the findings hold when using other datasets and models.

Additionally, we did not investigate whether the models' feature importance may change when using different encoding methods. This may be the case when the encoding methods learn that demographic features are not necessary for the prediction. However, given the correlation between demographic features and the target, it is unlikely that different encodings lead to models not contributing importance to demographic features at all. Still, this should also be investigated in the future.

Because it is not the major focus of our study, we have not investigated the relationship between the importance of activity- and performance-related features. Future research could investigate what is more important and how the two feature subsets relate to each other.

8. CONCLUSION

Our analyses show strong evidence that demographic features do not increase a model's performance on at-risk prediction as long as study-related information is available. Nonetheless, both our theoretic considerations, as well as our empirical evaluations indicate that demographic features correlate both with study-related features and the target. Thus, they are used by the models for the prediction, although this would not be necessary, leading to biases and, as a result, unfairness. Because of these fairness concerns, we advise leaving out demographic features and features pointing towards demographic characteristics. This should also make it possible to share more data between researchers, as it reduces privacy concerns. Nonetheless, our paper also shows that investigating the causal mechanisms of how demographic features impact academic achievement is worthwhile and should be encouraged. Deployments of at-risk prediction models should not include demographic features, though.

9. REFERENCES

[1] A. Al-Zawqari and G. Vandersteen. Investigating the role of demographics in predicting high achieving students. In *International Conference on Artificial Intelligence in Education*, pages 440–443. Springer, 2022.

[2] S. Alturki, I. Hulpuș, and H. Stuckenschmidt. Predicting academic outcomes: A survey from 2007 till 2018. *Technology, Knowledge and Learning*, pages 1–33, 2020.

[3] E. A. Amrieh, T. Hamtini, and I. Aljarah. Mining educational data to predict student's academic performance using ensemble methods. *International Journal of Database Theory and Application*, 9(8):119–136, 2016.

[4] S. Batool, J. Rashid, M. W. Nisar, J. Kim, T. Mahmood, and A. Hussain. A random forest students' performance prediction (rfspp) model based on students' demographic features. In *2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*, pages 1–4. IEEE, 2021.

[5] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox. Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, 8(1):014008, 2015.

[6] P. Cortez and A. M. G. Silva. Using data mining to predict secondary school student performance. 2008.

[7] A. Daud, N. R. Aljohani, R. A. Abbasi, M. D. Lytras, F. Abbas, and J. S. Alowibdi. Predicting student performance using advanced learning analytics. In *Proceedings of the 26th international conference on world wide web companion*, pages 415–421, 2017.

[8] E. Delahoz-Dominguez, R. Zuluaga, and T. Fontalvo-Herrera. Dataset of academic performance

evolution for engineering students. *Data in brief*, 30:105537, 2020.

[9] G. Fenu, R. Galici, and M. Marras. Experts' view on challenges and needs for fairness in artificial intelligence for education. In *International Conference on Artificial Intelligence in Education*, pages 243–255. Springer, 2022.

[10] L. Grinsztajn, E. Oyallon, and G. Varoquaux. Why do tree-based models still outperform deep learning on tabular data? *arXiv preprint arXiv:2207.08815*, 2022.

[11] P. K. Hee and L. Shuhan. Influences of economic capital, cultural capital and social capital on asian high school students' academic achievement. *Journal of Educational and Social Research*, 12(3):1–1, 2022.

[12] A.-S. Hoffait and M. Schyns. Early detection of university students with potential difficulties. *Decision Support Systems*, 101:1–11, 2017.

[13] Q. Hu and H. Rangwala. Towards fair educational data mining: A case study on detecting at-risk students. *International Educational Data Mining Society*, 2020.

[14] N. I. Jha, I. Ghergulescu, and A.-N. Moldovan. Oulad mooc dropout and result prediction using ensemble, deep learning and regression techniques. In *CSEDU (2)*, pages 154–164, 2019.

[15] A. U. Khasanah et al. A comparative study to predict student's performance using educational data mining techniques. In *IOP Conference Series: Materials Science and Engineering*, volume 215, page 012036. IOP Publishing, 2017.

[16] J. Kuzilek, M. Hlosta, and Z. Zdrahal. Open university learning analytics dataset. *Scientific data*, 4(1):1–8, 2017.

[17] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

[18] D. Micci-Barreca. A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *ACM SIGKDD Explorations Newsletter*, 3(1):27–32, 2001.

[19] V. L. Miguéis, A. Freitas, P. J. Garcia, and A. Silva. Early segmentation of students according to their academic performance: A predictive modelling approach. *Decision Support Systems*, 115:36–51, 2018.

[20] E. W. Neblett Jr, C. L. Philip, C. D. Cogburn, and R. M. Sellers. African american adolescents' discrimination experiences and academic achievement: Racial socialization as a cultural compensatory and protective factor. *Journal of Black psychology*, 32(2):199–218, 2006.

[21] L. Paquette, J. Ocumpaugh, Z. Li, A. Andres, and R. Baker. Who's learning? using demographics in edm research. *Journal of Educational Data Mining*, 12(3):1–30, 2020.

[22] F. Pargent, F. Pfisterer, J. Thomas, and B. Bischl. Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features. *Computational Statistics*, pages 1–22, 2022.

[23] R. Pishghadam and R. Zabihi. Parental education and social and cultural capital in academic achievement. *International Journal of English Linguistics*, 1(2):50, 2011.

[24] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

[25] N. J. Salkind. *Encyclopedia of research design*, volume 1. sage, 2010.

[26] R. Shwartz-Ziv and A. Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

[27] F. Sigrist. Latent gaussian model boosting. *arXiv preprint arXiv:2105.08966*, 2021.

[28] F. Sigrist. Gaussian process boosting. *Journal of Machine Learning Research*, 23(232):1–46, 2022.

[29] L. Sweeney. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems*, 10(05):557–570, 2002.

[30] M. Sweeney, J. Lester, H. Rangwala, A. Johri, et al. Next-term student performance prediction: A recommender systems approach. *Journal of Educational Data Mining*, 8(1):22–51, 2016.

[31] N. Tomasevic, N. Gvozdenovic, and S. Vranes. An overview and comparison of supervised data mining techniques for student exam performance prediction. *Computers & education*, 143:103676, 2020.

[32] B. Trstenjak and D. Donko. Determining the impact of demographic features in predicting student success in croatia. In *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1222–1227. IEEE, 2014.

[33] R. Yu, H. Lee, and R. F. Kizilcec. Should college dropout prediction models include protected attributes? In *Proceedings of the eighth ACM conference on learning@ Scale*, pages 91–100, 2021.

[34] M. Zaffar, M. A. Hashmani, K. Savita, and S. S. H. Rizvi. A study of feature selection algorithms for predicting students academic performance. *International Journal of Advanced Computer Science and Applications*, 9(5), 2018.

[35] Y. Zhao, Q. Xu, M. Chen, and G. Weiss. Predicting student performance in a master's program in data science using admissions data. In *Educational Data Mining*, 2020.

Scalable and Equitable Math Problem Solving Strategy Prediction in Big Educational Data

Anup Shakya
University of Memphis
ashakya@memphis.edu

Vasile Rus
University of Memphis
vrus@memphis.edu

Deepak Venugopal
University of Memphis
dvngopal@memphis.edu

ABSTRACT

Understanding a student's problem-solving strategy can have a significant impact on effective math learning using Intelligent Tutoring Systems (ITSs) and Adaptive Instructional Systems (AISs). For instance, the ITS/AIS can better personalize itself to correct specific misconceptions that are indicated by incorrect strategies, specific problems can be designed to improve strategies and frustration can be minimized by adapting to a student's natural way of thinking rather than trying to fit a standard strategy for all. While it may be possible for human experts to identify strategies manually in classroom settings with sufficient student interaction, it is not possible to scale this up to big data. Therefore, we leverage advances in Machine Learning and AI methods to perform scalable strategy prediction that is also fair to students at all skill levels. Specifically, we develop an embedding called MVec where we learn a representation based on the mastery of students. We then cluster these embeddings with a non-parametric clustering method where we progressively learn clusters such that we group together instances that have approximately symmetrical strategies. The strategy prediction model is trained on instances sampled from these clusters. This ensures that we train the model over diverse strategies and also that strategies from a particular group do not bias the DNN model, thus allowing it to optimize its parameters over all groups. Using real world large-scale student interaction datasets from MATHia, we implement our approach using transformers and Node2Vec for learning the mastery embeddings and LSTMs for predicting strategies. We show that our approach can scale up to achieve high accuracy by training on a small sample of a large dataset and also has predictive equality, i.e., it can predict strategies equally well for learners at diverse skill levels.

Keywords

Intelligent Tutoring Systems, Strategy Prediction, Equity, Representation Learning, Skill Mastery, Non-parametric Clustering, Fairness, Transformers, LSTM, Symmetry

A. Shakya, V. Rus, and D. Venugopal. Scalable and equitable math problem solving strategy prediction in big educational data. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 137–148, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115669

1. INTRODUCTION

The recent pandemic has spurred a remarkable growth in virtual learning and with it, the necessity to develop learning technologies that are effective even in the absence of face-to-face instruction. To this end, Intelligent Tutoring Systems (ITSs) [25] and more broadly Adaptive Instructional systems (AISs) will play a key role in education since they can scale up personalized instruction to large and diverse student populations. However, to adapt to a student, an AIS should be able to understand the student's thinking process which can be challenging. For instance, if we consider math learning, students can solve the same problem using several different approaches or *strategies.* Understanding these strategies can help an ITS/AIS adapt more effectively [22]. For example, the type of strategy can reveal the expertise/knowledge of a student in a topic, incorrect strategies that indicate misconceptions can be corrected by the ITS, the student can be trained to change strategy based on the problem context, and students may be less frustrated if the ITS guides them towards strategies that are more naturally aligned to their thinking.

In math problem solving, a *strategy* is a sequence of actions/steps that the student performs to solve a problem. An example of 3 different strategies is shown in Fig. 1. Human tutors can recognize different strategies followed by students and utilize these in one-on-one instruction. For instance, if a student is a visual learner, then they can teach the student to solve problems through visual aids, or if the student prefers an analytical approach to solve the same problem, then they can modify their teaching accordingly. However, adapting this approach for ITSs is challenging, particularly since identifying problem-solving strategies through computational methods is a complex problem. Specifically, there may be several strategies that are similar/symmetric without being completely identical. An example is illustrated in Fig. 1 to show similar and dissimilar strategies. As shown here, 2 of the 3 strategies are not exactly identical but implement the same idea and are thus symmetrical. The third strategy is quite different and asymmetrical to the first two strategies. Further, there may be several strategies that may not be conventional approaches to problem-solving but are indicative of unique ways in which students think about problems. Thus, if we identify a new strategy based on matching them with a set of previously known strategies, this approach may not be very effective when we want to scale up to big educational data. While there have been several approaches to detect strategies including using model tracing [4] or sequence mining [34] methods, newer advances in deep neural networks (DNNs) can learn much more complex representations from large-scale data. Thus, leveraging such DNNs, we predict *novel* strategies more effectively.

Our goal in this paper is to develop a scalable and equitable model to predict strategies in math learning. Specifically, though DNN models are highly effective, they may tend to produce biased results. For instance, since most DNNs have a loss function that optimizes the overall loss, depending on the data distribution used during training, their results may be unfair to some sub-groups in the data. In our context, we want to avoid the model being unfairly biased where it can only identify strategies for certain student sub-groups. Specifically, we want to avoid *disparate mistreatment* [35] where the model accuracy is significantly different for different types of learners. In particular, learners may have disparity in their mastery or skill level which will influence their choice of strategy for a problem. For example, in Fig. 1, the third strategy shown in the figure is more sophisticated than the other two and the student who applies this strategy is likely to have greater mastery in the topic. Therefore, we want to ensure that our model can predict strategies equally well for learners at all skill levels. To do this, we use a sampling approach, where instead of training the DNN over the full dataset (which may contain biases), we modify the underlying data distribution. Specifically, we sample the data such that sub-groups in the data are equally well-represented. Thus, when the DNN is trained over these samples instead of the full dataset, the DNN is forced to optimize its loss over all sub-groups. In general, sampling is a well-known approach used to scale up complex DNNs while training the model from large datasets [6]. Further, it has been shown that in some cases using too much data can lead to poor generalization [17]. In our case, a naive sampling approach where we sample students uniformly at random and train over strategies used by the sampled students will certainly be biased towards the skill level of the majority group and does not account for inequalities in skill levels. Therefore, here, we develop an iterative non-parametric clustering method where we cluster the data into groups where each group corresponds to strategies corresponding to similar skills levels. Further, since strategies themselves are hard to compare exactly, we develop an approach where we use *approximate symmetries* to group strategies. We then train a DNN to predict a strategy by sampling from these diverse groups.

We implement our approach using the *DP-Means* Hierarchical Dirichlet Process framework [9] to jointly cluster students and problems. Specifically, we project students (and problems) into an embedding space that we term *MVec* (Mastery Vectorization). To do this, we represent relationships between symbolic objects (students, problems, and concepts used in strategy) as a graphical structure. We then learn dense vectors using an embedding approach called *Node2Vec* [5] that assigns similar embeddings to nodes that have similar neighborhoods. We add mastery over concepts used in the strategy as weights in the graph estimated from a transformer model with attentions [31]. Thus, students with mastery over similar concepts in their strategies are assigned similar embeddings. We optimize the clusters incrementally where in each step, we adaptively change a penalty parameter based on the symmetries encoded by the clusters in the previous step. To quantify approximate symmetries, we develop a strategy alignment procedure with *positional encodings* [28]. Once the clusters converge, we sample training instances from the clusters and train a Long Short Term Memory (LSTM) model that predicts strategies.

We evaluate our approach on two datasets from MATHia, a commercial AIS widely used for math learning in schools. The data is available through the PSLC datashop [30]. The datasets are both large datasets that consist of millions of data instances (an instance is a student-problem pair and has multiple interactions in the dataset). Our results confirm that using our approach, we can sample a substantially smaller set of instances from the big dataset which we can use to train the strategy prediction model efficiently and achieve high accuracy in strategy prediction for students at diverse levels of mastery.

2. BACKGROUND

2.1 Related Work

Ritter et al. [22] provide a comprehensive survey on different approaches used to identify student strategies. Well-known approaches include the use of model tracing-based methods [4] to identify strategies. In such cases, strategies may be pre-specified and the tutor can recognize correct and incorrect strategies. Model-tracing-based methods have also been adapted to recognize new strategies [21]. Sequence learning approaches have been used in Open-Ended Learning Environments such as Betty's brain [11]. In [34], sequence pattern mining was applied to a MOOCs platform to analyze activity sequences of learners. For conversational tutors, natural language conversation interactions between tutors and students were mapped into a taxonomy of higher-level pedagogical concepts (e.g. scaffolding) by education experts [15]. These concepts can also be seen as a form of strategy and models have been developed to predict these concepts from conversational tutors [13, 26, 32]. Shakya et al. [27] developed an approach using importance sampling to sample data instances to scale up training of a strategy prediction model based on student interaction data from Mathia. Specifically, they formulated a Neuro-Symbolic AI model [33] where symbolic formulas were used in conjunction with a DNN to train the model. However, unlike our approach [27] has two fundamental limitations in identifying strategies. Particularly, their work does not use mastery to diversify the training samples which is important for equitable training. Further, it does not learn approximately symmetrical groups in a non-parametric manner. Thus, it cannot effectively group together symmetrical strategies which is necessary if we want to train the DNN from strategies that represent all such groups.

Mastery-based learning was proposed in the classic work by Bloom [1] to reduce achievement gaps between diverse students. The famous Bloom 2-sigma rule illustrates the benefits of such mastery-based learning. Ritter et al. [23] more recently provides a detailed insight into how mastery learning works in large-scale environments through their experiments on the MATHia platform. Knowledge tracing [4] is a well-known approach for inferring the *knowledge state* of students over KCs which indicates the degree of mastery over the KCs. More recently, deep knowledge tracing [20]

Solve linear equations: $5x - 2y = 4$ ①
$3x + 2y = 12$ ②

Strategy 1			Add ① and ②	Eliminate y	Collect x on left	Divide by coef. of x	find x	substitute x to find y
Strategy 2	Mul ① by 3	Mul ② by 5	Sub ① from ②	Eliminate x	Collect y on left	Divide by coef. of y	find y	substitute y to find x
Strategy 3			Formulate mat. mul. $AX = B$	Identify A, X and B	Compute A^{-1}	Compute $X = A^{-1}B$	find x and y	

Figure 1: Illustrating symmetries in strategies where similar colors indicate similar steps. Strategies 1 and 2 are similar in that they use the elimination method but are not identical. Strategy 3 uses the matrix method which indicates a higher level of sophistication in student mastery.

performed knowledge tracing using deep learning models. There is also a significant momentum in tackling the Knowledge Tracing problem in terms of graphs with the advent of GNNs [12, 16]. In [29], node-level and graph-level GCNs have been used to learn exercise-to-exercise and concept-to-concept relational sub-graphs adding to the semantic value of the representations. The natural phenomenon of learning, forgetting and dynamic changes to a student's mastery of knowledge concepts is formulated using gating-controlled mechanisms in [36]. Learning the pre-requisite structure of various associated skills has proven to be insightful to understand the problem-solving patterns [3, 19]. In [18], an attention-based model was proposed to predict correct answers but this was not used to predict strategies which is the focus of our work.

Our approach to using symmetries to make deep learning more scalable is inspired by the Geometric Deep Learning (GDL) [2] framework. Specifically, GDL is a formal framework used to understand the effectiveness of DNNs from the perspective of symmetries. Here, we ground GDL in the context of improving the effectiveness of DNNs in strategy prediction from big, diverse data. More generally, being selective about training instances has been shown to improve scalability and generalization [17]. Deep importance sampling [6] has the same underlying principle as our approach in that they propose to sample data to scale up training. However, unlike our approach they do not use symmetries as a basis for efficiently and equitably training the model. More recently, there has been work on improving fairness in DNNs by adaptively selecting batches during training to improve fairness measures such as minimizing gender disparity [24]. In principle, our approach also tries to achieve a similar goal in the context of educational data which is more challenging given that both mastery and strategies are complex variables.

2.2 Overview of Embedding Models

We use the well-known embedding model Node2Vec [5] to learn our mastery-based embedding MVec. Node2Vec is an embedding model for graphs and learns embeddings/dense vectors for nodes in the graph base on local neighborhoods. It is well-known to be a highly scalable approach for learning embeddings from large graphs. Node2Vec assigns similar vector representations for nodes with similar neighborhoods. Internally, it uses a skip-gram model called Word2Vec [14] to learn these representations. Word2Vec, which was originally developed for word embeddings, is used to predict neighboring nodes (also called context) from a given node. An autoencoder architecture is used in Word2Vec and the hidden layer learns the embedding. When neighborhoods are similar for two nodes, since their contexts are similar, the embedding learned for the two nodes will also be similar. Thus, Word2Vec projects the nodes into a continuous embedding space where similar/symmetrical nodes lie close to each other in the space.

2.3 DP-Means

DP-Means [10] is a non-parametric clustering algorithm that does not require us to specify of the number of clusters. The DP-Means Hard Gaussian Processes (HDP) clustering learns a 2-step hierarchy where *local clusters* for multiple datasets are learned at the lower level and these clusters are associated with *global clusters* at the higher level. Let x_{ij} denote the i-th instance of dataset j. The specific objective function of HDP is as follows.

$$\sum_{p=1}^{g} \sum_{x_{ij} \in \ell_p} ||x_{ij} - \mu_p||_2^2 + \lambda_\ell k + \lambda_g g \tag{1}$$

where ℓ_p is the p-th global cluster, k is the total number of local clusters, μ_p is the center of the p-th global cluster, g is the total number of global clusters, λ_ℓ is a local penalty that controls the formation of local clusters and λ_g is a global penalty that controls the formation of global clusters.

We can minimize the objective in Eq. (1) HDP clustering as follows. For each x_{ij}, we compute the distance to the current global cluster means. If the minimal distance exceeds $\lambda_\ell + \lambda_g$, we create a new local cluster for x_{ij} and a new global cluster ℓ_g associating it with the newly created local cluster. If the minimal distance is smaller than the sum of penalties, then we find the closest global cluster for x_{ij}, say $\ell_{g'}$. We then add x_{ij} to a local cluster that is already a part of $\ell_{g'}$. If no such local clusters exist, we create a new one for x_{ij} and associate it with $\ell_{g'}$. We then process the local clusters as follows. Let c denote a local cluster. We compute the global cluster whose mean is at a minimal distance, d' from c. Let the sum of distances of the points in the local cluster c to its cluster center be m. If d' is greater than the sum of the global cluster penalty and m, we create a new global cluster and assign c to this new global cluster. This algorithm converges to a locally optimal solution for Eq. (1) as shown in [10].

2.4 Positional Encodings

Positional encodings [31] are used to encode positional information in a sequence using a continuous vector space. Specifically, using sine and cosine functions that alternate with frequencies, we can represent positions in a sequence as follows. Let the position of the t-th item in the sequence be encoded by the d dimensional vector $\vec{p}_t$. The k-th dimension in $\vec{p}_t$ is computed as follows. If k is even, the value is equal to the sinusoidal function $sin(\omega_k.t)$ and if k is odd, the value is equal to the cosine function $cos(\omega_k.t)$, where $\omega_k = 1/10000^{2k/d}$. The frequencies of the sine and cosine functions increase as k increases. Positional encodings are widely used to augment the latent representation learned by a deep network with positional information for sequence learning.

3. PROPOSED APPROACH

Since strategy is a generic term, we define it more precisely. Specifically, we consider strategies in the context of structured interaction between students and tutors. In this case, a student interacts with a tutor and solves a problem by sequentially solving the steps that lead to the final solution. Thus, we can think of a strategy as a sequence of actions the student takes among possible sequences in an action-space. Operationally, each step in the sequence is associated with a specific *knowledge component* (KC) [8] which is defined by domain experts and corresponds to the concept/knowledge required to solve that step. Thus, in our discussion, a strategy corresponds to a sequence of KCs. Further, note that a step can be associated with multiple KCs in which case, we can just unroll the step to ensure that each step has a single KC. While it is possible to adapt our approach to perform structure prediction where instead of a single KC, a step can be mapped to a more complex structure (e.g. a graph), we leave this for future work and focus on the case where a single KC is mapped to a step in the strategy.

In this paper, the task that we want to solve is the following. Given a student s and a problem p, we predict the sequence of KCs that s will use to solve p. In particular, we assume that we have a large dataset $\mathcal{D}$ where we refer to an *instance* in the dataset as a pair $(s, p) \in \mathcal{D}$. We want to sample instances from $\mathcal{D}$ to train a model that takes as input $(s, p) \in \mathcal{D}$ and predicts strategies, i.e., variable-length sequences of KCs. We also assume that $\mathcal{D}$ contains correctness associated with each step in the strategies. Specifically, for an input (s, p), for each step that s takes to solve p, we know if s was successful in solving that step correctly. We use this information to determine the mastery of a student and based on this, we develop an embedding (vector representation) for students and problems. We then jointly cluster the embeddings using a non-parametric approach such that instances where the strategies are approximately symmetric are clustered together. Finally, we train an LSTM model to predict strategies by sampling the clusters. In the subsequent subsections, we first describe our embedding called MVec. Next, we apply DP-Means HDP clustering [10] to the embeddings while also incorporating approximate symmetries in strategies.

3.1 MVec Embeddings

To learn the MVec embedding, we use an approach that is similar to Node2Vec [5]. Specifically, we construct a relational graph $\mathcal{G} = (\mathbf{V},\mathbf{E})$ as follows. Each student, problem, and KC in the training data is represented as a node $V \in \mathbf{V}$. For every student S who uses KC K as a step to solve problem P, there exist 2 edges $E, E' \in \mathbf{E}$, where E connects the node representing the student to the node representing the KC and E' connects the node representing the KC to the node representing the problem. An example graph over 3 students, problems, and KCs is shown in Fig. 2. We now sample paths in the graph and learn embeddings for these paths using word embedding models (Word2Vec) [14]. Specifically, the objective function is as follows.

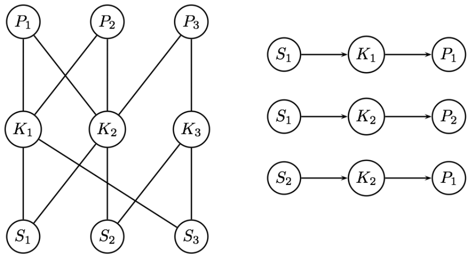


Figure 2: Illustrating a graph network of three students, problems, and KCs. The figure on the right shows some of the sampled random walks/paths.

$$\max_f \sum_{u \in \mathbf{V}} logP(N_Q(u)|f(u)) \tag{2}$$

where $f : u \rightarrow \mathbb{R}^d$ is the vector representation for nodes $u \in \mathbf{V}$, $N_Q(u)$ denotes the neighbors of u sampled from a distribution Q. Similar to Node2Vec, we assume that there is a factorized model that gives us a conditional likelihood that is identical to the likelihood function used in Word2Vec.

$$P(n_i|f(u)) = \frac{\exp(f(n_i) \cdot f(u))}{\sum_{v \in \mathbf{V}} \exp(f(v) \cdot f(u))} \tag{3}$$

where n_i is a neighbor of u. The conditional likelihood is optimized by predicting neighbors of u using u as input in an autoencoder neural network. The hidden layer learns similar embeddings for nodes with symmetrical neighborhoods. To do this, we generate walks on $\mathcal{G}$ as shown for the example in Fig. 2, and in each walk, given a node, we predict neighboring nodes similar to predicting neighboring words in sentences. To generate these walks, a simple sampling strategy Q is to randomly sample a neighbor for a node. However, in our case, it turns out that each neighbor may have different importance when it comes to determining symmetry. Specifically, if a student has achieved mastery in applying a KC to a problem, then the corresponding edges should be given greater importance when determining symmetry between nodes in $\mathcal{G}$. To do this, we train a Sequence-to-Sequence attention model [31] from which we estimate the sampling probabilities for edges in $\mathcal{G}$.

The intuitive idea in quantifying mastery is illustrated in Fig. 3 which shows the opportunities given to 3 students to apply KCs in different problems. For each sequence of KCs, we predict if the student got the step correct or wrong on the first attempt (abbreviated as CFA for Correct First Attempt) when given an opportunity to apply the KC. The CFA values are performance indicators for the student, i.e., if they have mastered a KC, then they are likely to get the step correct in every opportunity they get to apply that KC. We train a model to predict the CFA values (CFA = 1 indicates a correct application of the KC) given the KCs used in a problem. The predicted values from the model are shown for each KC. The bar graphs show mastery over the KCs. As seen here, the first student is inconsistent in applying

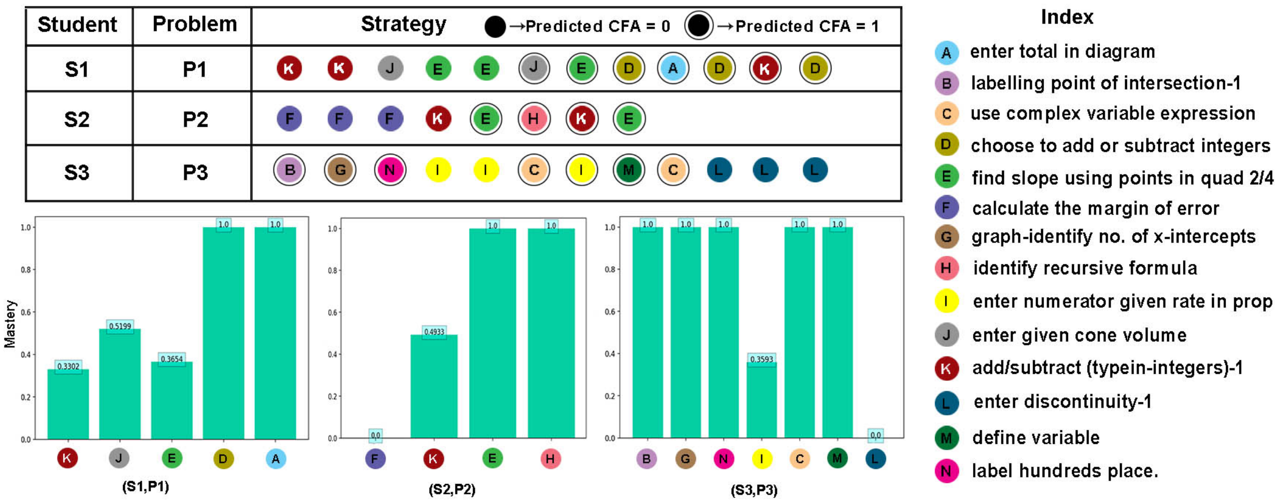


Figure 3: An example to illustrate the use of attention for mastery estimation. The bar charts show for each KC, the attention on a KC across steps that the student solves successfully (CFA=1) normalized by total attention for that KC. Larger values indicate that the model believes the student understands the KC as the attention on it is large when CFA=1 and vice versa.

the skill, *find the slope using points* (labeled as E) since the predictions for this oscillate between 0 and 1 whenever the student tries to apply this KC. On the other hand, student 2 consistently applies the same skill correctly and therefore the attention value is higher. We train the attention model from opportunities based on curriculum structure. Specifically, the curriculum consists of multiple units and each unit is further subdivided into sections. For each student S, from every unit that the student has completed say U, we select a problem P from each section that the student has worked on in U and train the model to predict the CFA values for each KC used in P. We use the standard architecture described in [31] for this model. Specifically, the input consists of the KC sequence, and the encoder maps this sequence to a latent representation and the decoder decodes the CFA values one at a time. The attention is given by

$$Attention(\gamma, \kappa, \eta) = softmax\left(\frac{\gamma\kappa^T}{\sqrt{d_k}}\right)\eta \qquad (4)$$

where γ, κ, and η are the standard query, key, and value matrices respectively as defined in [31], and d_k is the dimensionality of the embedding that represents the latent representations. We use the encoder-decoder attention, i.e., the query is the decoder representation and the key is the encoder representation. The attention weights are an estimate of the alignment between encoded latent representations of mastery with the decoded representation of correctly applying a skill at each step in the problem. The projection of mastery over a KC K based on the attention vectors is estimated by the following equation.

$$\alpha(S, P, K) = \frac{\sum_i \sum_{v \in \pi(a_i)} v}{\sum_i \sum_{v \in \pi(a_i)} v + \sum_i \sum_{v' \in \bar{\pi}(a_i)} v'} \qquad (5)$$

where $\pi(\cdot)$ extracts only those values in the input vector where the corresponding output for that step is predicted as 1, i.e., the model predicted that the student could solve the step correctly. $\bar{\pi}(\cdot)$ is the complement of $\pi(\cdot)$, i.e., it extracts attention values corresponding to steps that were predicted as mistakes made by the student and i sums up all the instances where K is used.

We now sample paths from $\mathcal{G}$ using the factored distribution, i.e., $Q(S) * Q(K|S) * Q(P|K, S)$, where $Q(S)$ is the probability of sampling a student node, $Q(K|S)$ is the probability of sampling a KC K given student S and $Q(P|K, S)$ is the probability of sampling problem P given K, S. We assume that $Q(S)$ is a uniform distribution over students. The conditional distributions are as follows.

$$Q(K|S) = 1/n \sum_p \alpha(S, P, K) \qquad (6)$$

$$Q(P|K, S) = \alpha(S, P, K) \qquad (7)$$

where n is the number of opportunities given to student S to apply KC K. The algorithm to generate MVec embeddings is shown in Algorithm 1. As shown here, we sample a path in the graph as follows. We first sample student S uniformly at random, then we sample a KC K from $Q(K|S)$ and a problem from $Q(P|K, S)$. We then predict each node in the path using the neighboring nodes through a standard Word2Vec model. The resulting embeddings are learned in the hidden layer of the Word2Vec model. Note that for scalability, we do not construct/store the full graph $\mathcal{G}$ at any point. Instead, we only sample paths in an online manner as shown in Algorithm 1.

3.2 Non-Parametric Clustering

We cluster the student and problem MVec embeddings jointly through a non-parametric approach based on symmetries defined as follows. For the dataset denoted by $\mathcal{D}$, let $\mathbf{S}$, $\mathbf{P}$ denote the set of students and problems respectively in $\mathcal{D}$.

Definition 1. *A* strategy-invariant *partitioning w.r.t $\mathcal{D}$ is a partitioning $\{\mathbf{S}_i\}_{i=1}^{k_1}$ and $\{\mathbf{P}_j\}_{j=1}^{k_2}$ such that $\forall i, j$, if $S, S' \in \mathbf{S}_i$, $P, P' \in \mathbf{P}_j$, S, S' follow equivalent strategies for P, P' respectively.*

where k_1 and k_2 are the number of partitions/clusters for

Algorithm 1 Generate MVec embeddings

```
Input: Relation Graph: $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ with student, problem
    and KCs as nodes, Embedding dimension: $d$, pre-trained
    attention-model $\mathcal{A}$
Output: Embeddings for each node $v \in \mathbb{R}^d$
    Initialize: set of walks, $\mathcal{W} = empty$
for all $t = 1$ to $T$ do
    Sample a path $< S, K, P >$ in $\mathcal{G}$ from $Q(S) * Q(K|S) * Q(P|K, S)$ using Eq. (6) and (7).
    $\mathcal{W} = \mathcal{W} \cup < S, K, P >$
end for
$v_e = word2vec(\mathcal{W}, d)$
return $v_e$
```

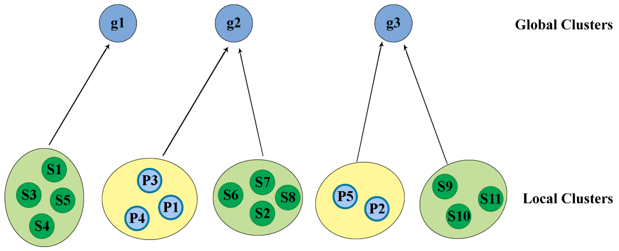


Figure 4: HDP Clustering showing the local clusters (student clusters and problem clusters) and the global clusters that combine the student, problem clusters.

students and problems respectively. The benefit of strategy-invariant partitioning is that we can scale up without sacrificing accuracy by training a prediction model only on samples drawn from the partitions instead of the full training data. Therefore, our task is to learn such partitioning approximately (since constraining the partitions to have exact equivalence of strategies is a hard problem). Since it is hard to know apriori how many partitions are needed, we formulate this as a non-parametric clustering problem and use DP-Means [10] to learn the clusters.

To formalize our approach, we begin with some notation. Let $\mathbf{S} = \{x_{i1}\}_{i=1}^{N}$ denote the set of students and $\mathbf{P} = \{x_{j2}\}_{j=1}^{M}$ denote the set of problems. We refer to the student and problem clusters as the *local* clusters. A *global* cluster combines student and problem clusters as illustrated in Fig. 4. We run the standard DP-Means HDP clustering algorithm to optimize Eq. (1) and learn global clusters that combine local clusters over $\mathbf{S}$ and $\mathbf{P}$. Note that large values of the global penalty λ_g result in a coarse clustering with few clusters and small values of the penalty result in fine-grained clusters. We adaptively change λ_g where we progressively lower the penalty yielding a *coarse-to-fine* refinement of the clusters. Specifically, suppose $\ell_1 \dots \ell_g$ are the current global clusters, we compute a score $\mathcal{S}(\ell_1 \dots \ell_g)$ based on the symmetry of strategies within each cluster and as long as the score progressively improves across iterations, we reduce λ_g to obtain finer-grained clusters.

3.3 Refining Clusters using Symmetry

Note that each global cluster implicitly represents a set of strategies, i.e., a student-problem pair (s, p) within the cluster corresponds to a strategy followed by s for problem p.

Algorithm 2 Coarse-to-Fine Refinement

```
Input: Student/Problem set: $\{x_{ij}\}$, Constant Penalty parameter: $\lambda_\ell$, iteration limit $T$
Output: Global strategy clustering $\{\ell_1, \dots, \ell_g\}$
    Initialize : Global cluster penalty $\lambda_g = y$ (where $y$ is a
    large number), $t = 0$, cluster coherence $coh_{t-1} = 0$.
repeat
    $t = t + 1$
    Cluster with penalties $\lambda_\ell$, $\lambda_g$
    Compute cluster coherence score, $coh_t = \mathcal{S}(\ell_1, \dots \ell_g)$
    Reduce: $\lambda_g = \lambda_g - \epsilon$
until $coh_t > coh_{t-1}$ or $t > T$
```

We want to quantify symmetry between strategies within a cluster. A naive approach to compare two strategies is to compute the mean of the MVec embeddings for the KCs used in each strategy and then compute the distance between the means. However, this assumes that all permutations of a strategy are equivalent to each other which is problematic. On the other hand, suppose we compare the KC embedding at a step in one strategy with the KC embedding at the same step in the other strategy, then we assume the strategies are equivalent only if they are perfectly aligned with each other which is also an over-simplification.

To match strategies approximately, we represent a strategy using a combination of embeddings and positional encodings [31], and approximately align two strategies to estimate the symmetry between them. A KC K in the strategy is represented by its positional embedding $\vec{K} = \vec{K_e} + \vec{K_p}$ where $\vec{K_e}$ is the MVec embedding for K and $\vec{K_p}$ is the positional encoding for K in the strategy. To compute symmetry between strategies, we compute an alignment between their positional embeddings. Alignment is a fundamental problem in domains such as Bioinformatics where a classical approach that is often used is the Smith and Waterman algorithm (SW) [28]. The idea is to perform local search to compute the best possible alignment between two sequences. SW requires a *similarity function* which in our case is the similarity between two KCs and we set this to be $s(K, K')$ $= \vec{K}^\top \vec{K}'$, i.e., the cosine similarity between the positional embeddings of K and K'. Further, SW also requires a *gap penalty* which refers to the cost of leaving a gap in the alignment. In our case, we set the gap penalty to 0 since we want symmetry between strategies to be invariant to gaps. That is, if two strategies are symmetric, adding extra steps in the strategies is acceptable. SW iteratively computes a scoring matrix based on local alignments. The worst-case complexity to compute the scoring matrix that gives us scores for the best alignment is equal to $O(m * n)$ where m and n are lengths of the strategies.

Note that in our case, we are interested in quantifying symmetry between strategies based on the alignment. Specifically, let $\mathbf{K}$ and $\mathbf{K}'$ be two strategies of lengths n and m respectively. SW gives us an alignment between $\mathbf{K}$ and $\mathbf{K}'$ denoted by $L(\mathbf{K}, \mathbf{K}')$. The alignment consists of the pairs KCs from $\mathbf{K}$ and $\mathbf{K}'$ respectively that have been matched/aligned or a gap, i.e., a KC from $\mathbf{K}$ could not be aligned with any KC from $\mathbf{K}'$. We compute the symmetry score between $\mathbf{K}$ and $\mathbf{K}'$ as $r(\mathbf{K}, \mathbf{K}') = \frac{1}{\max(n,m)} \sum_{(K,K') \in L(\mathbf{K}, \mathbf{K}')} (\vec{K}^\top \vec{K}')$, where

$(K, K') \in L(\mathbf{K}, \mathbf{K}')$ are aligned KCs and $\vec{K}^\top \vec{K'}$ is their cosine similarity. We see that $0 \leq r(\mathbf{K}, \mathbf{K}') \leq 1$. Based on this, we estimate symmetry in the clustering as follows.

$$\mathcal{S}(\ell_1, \ldots, \ell_g) = \frac{1}{g} \sum_{p=1}^{g} \mathbb{Z}_p \sum_{\mathbf{K}, \mathbf{K}' \in T(\ell_p)} r(\mathbf{K}, \mathbf{K}') \qquad (8)$$

$T(\ell_p)$ is a set of all strategies in ℓ_p and $\mathbb{Z}_p = \frac{2}{|T(\ell_p)|(|T(\ell_p)|-1)}$ is the normalization term. Thus, a larger value of $\mathcal{S}(\ell_1, \ldots, \ell_g)$ implies that the clustering corresponding to $\ell_1, \ldots, \ell_g$ has a greater degree of symmetry in strategies. Using this score, we refine the clustering by adapting the global penalty. Specifically, we reduce the global penalty λ_g by a constant ϵ as long as the symmetry score decreases across iterations or for a fixed number of iterations. Algorithm 2 summarizes the coarse-to-fine refinement.

3.4 Training the Model

We use an LSTM architecture similar to [27] to predict strategies. Specifically, the model is a one-to-many LSTM that takes student, problem vectors as input and generates a sequence of KCs as output. To train this model, we sample instances from the converged global clusters, and for each sampled student-problem pair (s, p), the LSTM input is the concatenation of MVec embeddings of s and p. The output corresponds to the sequence of KCs in the strategy used by s for p, each of which is encoded as a one-hot vector. To handle variable-length strategies, a special *stop* symbol is used to denote the end of a sequence. The entire model is trained using the standard categorical cross-entropy loss.

4. EXPERIMENTS

Our goal is to answer the following questions through our evaluation. i) what is the accuracy of our approach in predicting strategies? ii) how does our approach scale-up? iii) what is the influence of mastery in predicting strategies accurately? and iv) is there a disparity in the accuracy of prediction for different skill-based sub-groups in the data?

4.1 Dataset

The data we use in this work is large-scale real-world education data recorded with real students using MATHia. MATHia is an online math learning program for middle school students that is popularly used across several schools. We used two datasets provided by MATHia for evaluating our proposed approach, Bridge-to-Algebra 2008-09 (`BA08`) and Carnegie Learning MATHia 2019-20 (`CL19`). Both of these datasets contain recorded interactions between the student and the computer tutor while the student attempts to solve a problem on the platform. Each recorded interaction consists of the log of the student's action toward solving the problem, for example, the knowledge component used, if hints were needed and if the step was completed correctly. `BA08` is an older dataset that consists about 20 million interactions for about 6000 students and $52k$ unique algebra problems. This dataset contains about 1.6 million *data instances*. It is important to note that we consider a *data instance* as a student-problem pair, so one *data instance* consolidates all the interactions/steps for one student on a specific problem. `CL19` is a more recent and larger dataset containing about 47 million interactions for 5000 students and about $32k$ unique math problems. It has about 1.9 million *data instances*. Both datasets are publicly available through the PSLC datashop [30].

Table 1: Main parameters for the models.

Transformer-based model	**LSTM-based strategy model**
Dimension → 512	Latent Dimension → 200
Number of layers → 6	Epochs → 60
Number of heads → 8	Batch Size → 30
Dimensions of key, value and query → 64	Adam Optimizer with Learning rate 0.01
Max Sequence Length → 150	Dropout → 0.1
Dropout → 0.1	
Weight Sharing → False	

4.2 Experimental Setup

To train the attention model, we used the transformer implementation in [31]. For the strategy prediction, we used a one-to-many LSTM [27] where the input is the student and problem embedding, and the output is the sequence of KCs. The parameters for the two models are shown in Table 1. We used the standard parameters for the transformer model and retained the same parameters as in [27] for the LSTM model for an unbiased comparison. For generating MVec embeddings, we used Gensim [14] with an embedding dimension set to 300 (which is typically used). We initialize the local cluster penalty $\lambda_\ell = 7$ and global cluster penalty $\lambda_g = 9$ for Coarse-to-Fine refinement and reduce the global penalty by $\epsilon = 1$ (we discovered these to be the best-performing hyper-parameters in experimentation). We perform our experiments on a machine with 64 GB RAM, an Nvidia Quadro 5000 GPU with 16 GB memory, and a CPU with 8 cores. The code for our implementation is available here [1].

4.3 Comparison to Baselines

We compared our approach with the following methods. The first one is a specialized approach proposed in Shakya et. al. [27] (CS) for the same datasets where an LSTM is trained using importance sampling. However, this sampling does not incorporate mastery or approximate symmetries to find diverse training instances. We also applied a more general importance sampling approach that is said to be applicable for any DNN model training proposed in [7] (IS) using their publicly available implementation. However, IS failed to output any results for datasets of our size and therefore we do not show it in our result graphs. This indicates that general-purpose methods do not scale up for our datasets. We also developed a stratified sampler (GS for group sampling) where the distribution is only proportional to the number of problems solved by a student, i.e., we sample more instances from students that have data associated with them. The last baseline is a naive Random Sampler (RS) used as a validation check where we sample students and problems uniformly at random. We refer to our approach as Attention Sampling (AS). In our evaluation, for each approach, we enforce a limit on the number of training instances and measure test accuracy based on the model trained with this limit. This is similar to a measure of the *effective model complexity* [17] which is the number of training samples to achieve close to zero error. We report the average accuracy of predicted KCs based on three training runs.

[1] `https://github.com/anupshakya07/attn-scaling`

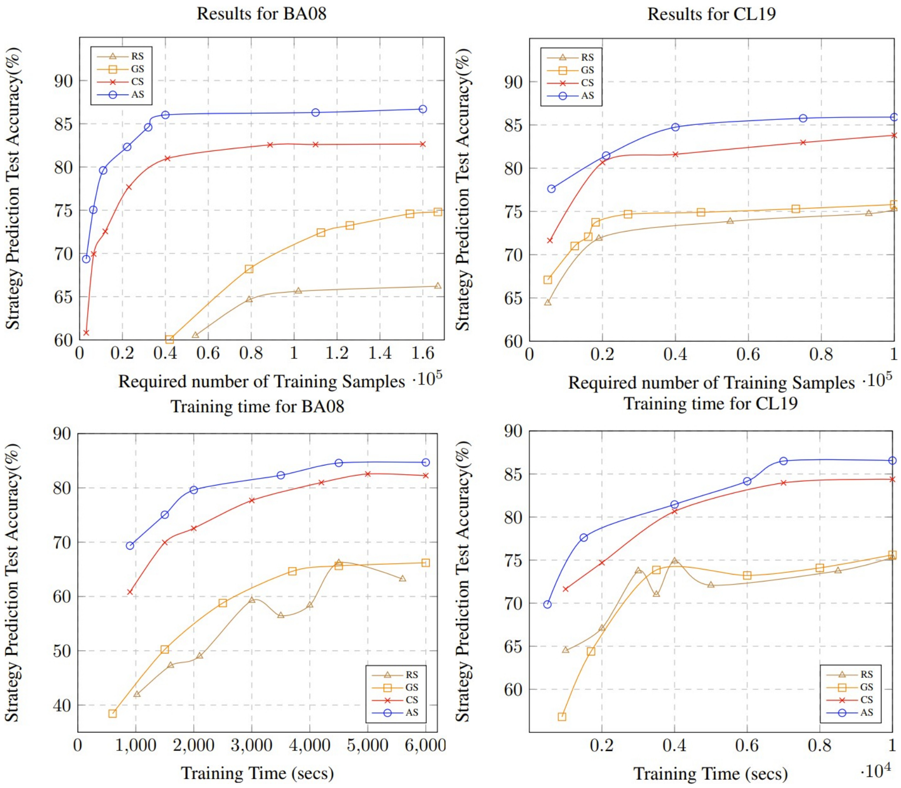


Figure 5: Illustrating Scalability vs Accuracy. (a), (b) show test accuracy for strategy prediction for varying training datasize limits. (c), (d) show accuracy (strategy prediction) for different training time limits.

4.4 Results and Discussion

4.4.1 Accuracy

The strategy prediction accuracy results for `BA08` and `CL19` are shown in Fig. 5 (a) and (b). As shown in Fig. 5 (a), for `BA08`, in our approach (AS), it takes less than 1% of the entire data (of `BA08` containing 1.6 million instances) to obtain test accuracy that is greater than 80%. CS is the next best performer but is consistently below AS for all training sizes. GS performs significantly worse which illustrates that symmetries are more complex and a simple grouping based on problems/students is insufficient. The poor performance of RS validates that the problem of choosing the correct samples is a challenging one. As seen in Fig. 5 (b), for a considerably larger dataset `CL19`, we can observe similar performance as in `BA08`. AS remains the best performer and here CS is less stable since we see a performance drop as we increase the limit on training samples. This suggests that CS may not be able to capture all symmetries and thus may produce a more biased training sample set. The results for GS and RS are similar to those observed in `BA08`. As mentioned before, IS failed to produce any results.

4.4.2 Scalability

Fig. 5 (c) and (d) show the training time required to obtain a specific accuracy for `BA08` and `CL19` respectively. Even with the extra processing that is needed to compute the mastery-based embeddings and the non-parametric clustering, AS requires the shortest training time to achieve an accuracy that is higher than the other approaches. This illustrates the significance of leveraging symmetries in the data to train the model. As mentioned before, the full data is infeasible to train and when attempting to use the full data, the model did not converge for both datasets even after several days of training time using our experimental setup. As seen in our results, for `CL19`, the training time is larger since it takes longer to compute the groups using non-parametric clustering due to the much larger size of the dataset. However, considering that `CL19` is significantly larger than `BA08`, we see that AS could still scale up to this dataset quite easily while IS which is a state-of-the-art sampling method for DNN training failed to train the model.

Table 2: Ablation study with NS (No symmetries used), SS (Symmetries without using mastery) and MS (Adding the mastery model to better identify symmetries). Results are shown for 2 datasets with different sample sizes. Accuracy results in %.

Expts.	**BA08**			**CL19**		
	40k	**100k**	**150k**	**40k**	**80k**	**100k**
NS	60.05	71.14	74.58	74.81	75.4	75.8
SS	80.98	82.3	82.65	81.6	83.2	83.8
SS + MS	86.02	86.21	86.53	84.74	85.8	85.9

4.4.3 Ablation Study

Table 2 shows the results of our ablation study. We add each component to our overall approach and observe the test accuracy as we vary the sample size in the training data. Specifically, the first case (**NS**) uses no symmetries,

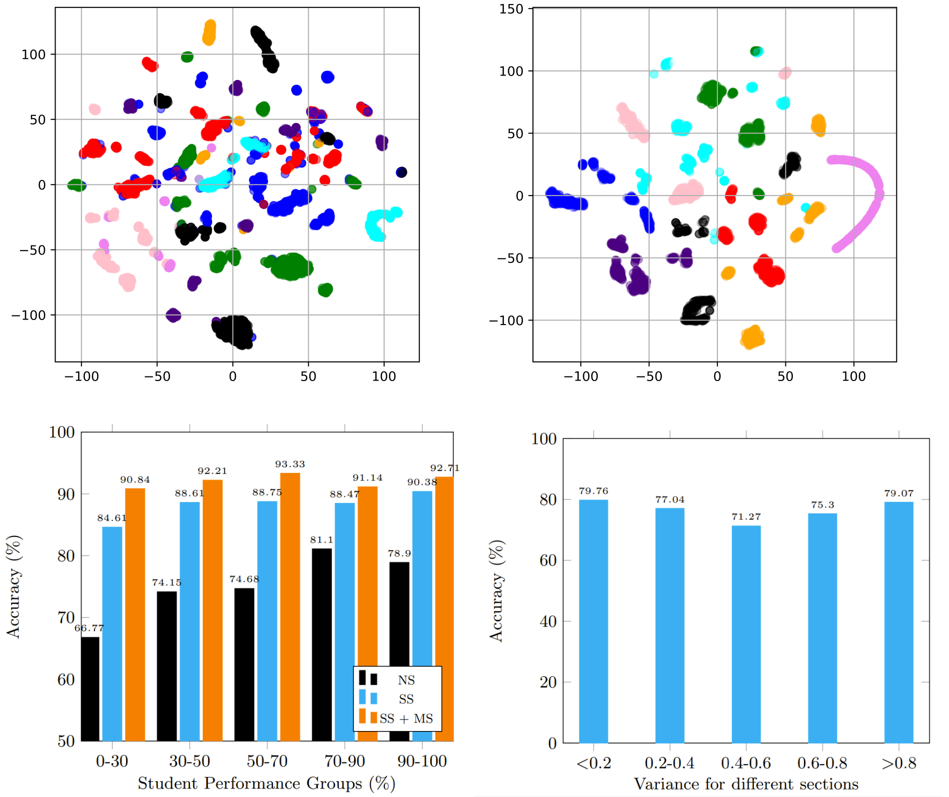


Figure 6: T-SNE visualization of strategy clusters for CL19. The color-coded plots show the 2D representation of the different strategy clusters for (a) Embeddings that do not use mastery (b) MVec embeddings. The strategy representations are extracted from the final hidden layer of the LSTM model and converted to 2D representation using T-SNE. (c) shows accuracy for different groups of students (based on their performance) for CL19. The x-axis denotes different ranges of %s, where a range $a - b$ denotes that students in this group got $> a$ and $< b$ steps correct in their first attempt. The y-axis shows accuracy over the groups. (d) shows the performance of the model on different groups based on the average variance of the strategies in the sections for CL19. Variance is computed using edit distance as the metric of similarity between strategies.

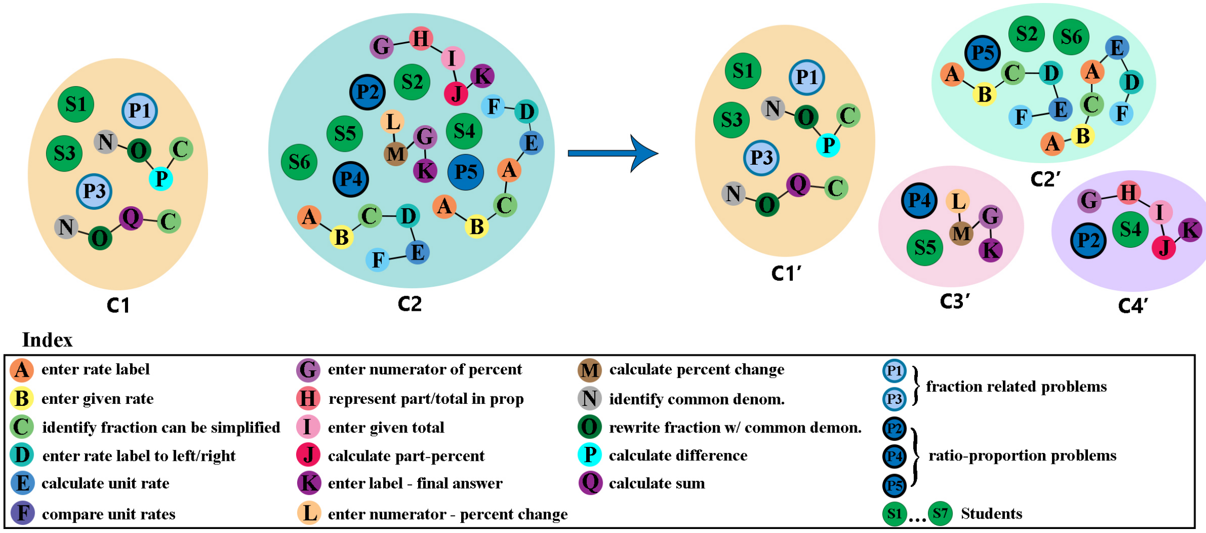


Figure 7: An example from the dataset CL19 illustrating coarse-to-fine refinement of clusters. Strategies are shown by paths connecting KCs. **C1** and **C2** are the coarse clusters which get refined into strategy invariant clusters **C1′**, **C2′**, **C3′** and **C4′**.

Table 3: Different strategies used by the students for different problems in the same section for CL19 dataset. The model is able to predict accurately as student adapt their strategies.

Student	Problem Name	Predicted Strategy	Actual Strategy
S_1	linear inequalities numberline 5	represent open point on numberline-1	represent open point on numberline-1
		represent ray on numberline-1	represent ray on numberline-1
		represent inequality in symbolic problem-1	represent inequality in symbolic problem-1
		identify when finished with numberline-1	identify when finished with numberline-1
		identify invisible non-inflection point is in solutionset-1	identify invisible non-inflection point is in solutionset-1
		identify invisible non-inflection point is not in solutionset-1	identify invisible non-inflection point is not in solutionset-1
		identify visible non-inflection point is not in solutionset-1	identify visible non-inflection point is not in solutionset-1
	linear inequalities numberline 9	write simple inequality in verbal problem-1	write simple inequality in verbal problem-1
		represent closedpoint on numberline-1	represent closedpoint on numberline-1
		represent ray on numberline-1	represent ray on numberline-1
		identify when finished with numberline-1	identify when finished with numberline-1
		identify visible non-inflection point is not in solution set-1	identify visible non-inflection point is not in solution set-1
		identify invisible non-inflection point is not in solution set-1	identify invisible non-inflection point is not in solution set-1
		identify inflection point in solution set-1	identify inflection point in solution set-1
			identify invisible non-inflection point is not in solution set-1
			identify inflection point in solution set-1
S_2	ratio proportion prop1 4	enter part in proportion with variable-1	enter part in proportion with variable-1
		enter given total in proportion-1	enter given total in proportion-1
		enter numerator of given rate in proportion-1	enter denominator of given rate in proportion-1
		enter denominator of given rate in proportion-1	enter numerator of given rate in proportion-1
		enter proportion label in numerator-1	enter proportion label in numerator-1
		enter proportion label in denominator-1	enter proportion label in denominator-1
		calculate part in proportion with fractions-1	calculate part in proportion with fractions-1
		enter numerator of form of 1-1	enter denominator of form of 1-1
		enter denominator of form of 1-1	enter numerator of form of 1-1
		enter calculated value of rate-1	enter calculated value of rate-1
	ratio proportion prop1 5	enter proportion label in numerator-1	enter proportion label in numerator-1
		enter proportion label in denominator-1	enter proportion label in denominator-1
		enter given total in proportion-1	enter given total in proportion-1
		enter numerator of given unit rate in proportion-1	enter numerator of given unit rate in proportion-1
		enter denominator of given unit rate in proportion-1	enter denominator of given unit rate in proportion-1
		calculate part in proportion with fractions-1	calculate part in proportion with fractions-1
		enter calculated value of rate-1	enter calculated value of rate-1

i.e., the clustering is performed randomly. Next, we cluster based on embeddings without using the mastery, i.e., when we generate the embeddings for MVec, we do not use the attention model and simply use triplets (S, P, K), where S is a student, P is a problem and K is a KC used by S for P as input to Word2Vec and generate embeddings. Thus, we use symmetries in strategy without utilizing mastery when we generate the clusters. We show this as Strategy Symmetry (**SS**) in the table. Finally, we add mastery to generate embeddings denoted by **SS** + **MS** and as shown, this improves the generalization performance for all sample-sizes thus, illustrating that utilizing mastery to learn embeddings plays a significant role in improving accuracy in predicting strategies.

4.4.4 Visualizing Clusters

We used T-SNE to visualize the clusters of strategies. For this, we pick 100 student-problem pairs sampled from 10 clusters. We then perform strategy prediction for these and visualize the hidden-layer representation of the LSTM in the T-SNE plot. We compare this for MVec embeddings as well as embeddings that are learned without using mastery. As shown in Fig. 6 (a) and (b), when we use MVec, the LSTM hidden-layer representation of strategies has better separation. This indicates that we learn better grouping of strategies using MVec embeddings.

4.4.5 Fairness

We evaluate if our approach results in disparate mistreatment. Specifically, this means that the model should not have significantly different accuracy for different sensitive sub-groups in the data. In our case, the sensitive sub-groups correspond to students at different skill levels. That is, we want to predict the strategies equally well for all students. To do this, we conducted an experiment where we divide the test data into 6 performance groups. The performance groups are based on the % of problem steps the students solve correctly on their first attempt. The performance groups include students who scored in the following ranges $\leq 30\%$, $30-50\%$, $50-70\%$, $70-90\%$, $\geq 90\%$. To measure disparate mistreatment, we compare the average accuracy of strategies predicted for each of these groups. For a student S in performance group $\mathbf{G}$, we predict the strategies for all problems attempted by S in the test set and measure the average accuracy μ_S. We then compute the accuracy over a performance group as $1/|\mathbf{G}| \sum_{S \in \mathbf{G}} \mu_S$. Fig. 6(c) shows our results for the variants, NS, SS and SS+MS (identical to those used in the ablation study) for CL19 (we show results on this since this is the larger and more recent dataset). As seen from our results, SS+MS yields the best accuracy over each performance group. Further, the accuracy over the poorest and the best performers is comparable to each other and not significantly different. Thus, there is no disparate mistreatment of any performance group shown by our approach.

Next, we want to verify if there is disparate mistreatment when we consider sub-groups that have rare strategies. To measure this, we divided the problem sections in the test set into groups based on the variance among strategies for problems in those sections. Specifically, to perform worst-case analysis, we used the edit distance to measure the variance of strategies within problems in a section. That is, if a pair of problems vary in two out of 10 steps, the edit distance is 0.2. We computed the variance in edit distances over all the problems in a section. We then obtained the sub-groups at 5 different thresholds of variance. Thus, groups that have large variance include more rare strategies, while groups that have smaller variance have fewer rare strategies. For all the problems in each of these sub-groups, we computed the average accuracy in strategy prediction. Fig.6(d) shows our accuracy results over all the sub-groups. As seen here, we have no disparate mistreatment for any of the sub-groups. Thus, we show that even in cases where rare strategies are used by students, our approach predicts strategies with an accuracy that is very similar to cases where common strategies are used.

4.4.6 Example Cases

Table 3 illustrates examples corresponding to two different students where we predict strategies for two problems taken from the same section in each case. Note that the students make modifications to their strategy to suit the problem context as seen in the examples, though the overall strategies are similar since the problems are from the same section. The model is able to successfully adapt and predict these strategy changes quite accurately. In the case of student S_1, for the second problem, the model predicted most of the steps except that the student had some redundant steps at the end which were not predicted by the model. In the case of S_2, for problem 1, the predicted strategy interchanged the order of a couple of steps that clearly does not significantly alter the underlying strategy.

We illustrate some examples of coarse-to-fine refinement in Fig. 7. Specifically, we show examples from two types of problems, *Fractions* and *Ratio, Proportions*. The clusters indicate the students, problems, and strategies followed by students. In cluster **C1**, even though there are two different strategies, they are symmetric to each other and therefore, in a subsequent iteration of refinement, $\mathbf{C1'}$ is the same as **C1**. On the other hand, **C2** consists of 4 strategies, 2 of these are expert-level strategies and the other two are simpler but differing strategies. Upon refinement of **C2**, we get $\mathbf{C2'}$ which intuitively represents the expert students and $\mathbf{C3'}$, $\mathbf{C4'}$ which represents students using simpler yet different strategies. Thus, the coarse-to-fine refinement results in invariant strategies within each cluster.

5. CONCLUSION

We presented a scalable and equitable framework for predicting math problem-solving strategies used by students. Since students with differing skill levels use significantly different strategies, to predict these, we need to train a model over diverse training instances. Identifying such instances is a challenging problem in big data. Particularly, identifying strategies which are approximately symmetrical to each other is a hard task. Here, we developed a clustering approach to discover diverse groups where instances within each group have approximately symmetrical strategies. Specifically, we learned an embedding MVec using a combination of Node2Vec where we learned representations for relationships in the data encoded as a graph and a transformer model that predicts mastery. Specifically, similar attentions in the transformer model over steps in the strategy indicated similar mastery in solving a problem, which we used to learn the Node2Vec representation. We then clustered the MVec embeddings with a non-parametric algorithm called DP-Means by iteratively refining the clusters based on the level of symmetry encoded within the clusters. By sampling from clusters, we were able to train an LSTM model to predict strategies using small but highly informative instances that were representative of strategies in the full data. Further, by sampling from clusters, we ensured that the LSTM model did not optimize its parameters for any specific group, but instead generalized over all groups in the data, thus making the model capable of identifying strategies from diverse groups. Experiments on two large-scale datasets demonstrated our accuracy in predicting strategies with a small fraction of the dataset and further, our predictions were fair across students at different levels of skill.

As part of future work, we hope to extend this model to non-structured interactions (e.g. conversations). Further, we also plan to explore more complex mappings of strategies where each step can be represented by a structure (e.g. graph, table, etc.) and developing structured prediction models from such mappings. We also propose to utilize this approach in instructional design where we can select problems to solve based on a student's predicted strategy and also to develop interventions in ITSs based on misconceptions identified in predicted strategies.

6. ACKNOWLEDGEMENT

This research was sponsored by the National Science Foundation under NSF IIS award #2008812 and NSF award #1934745. The opinions, findings, and results are solely the authors' and do not reflect those of the funding agencies.

7. REFERENCES

[1] B. S. Bloom. Learning for mastery. instruction and curriculum. regional education laboratory for the carolina and virginia, topical papers and reprints, number 1. *Evaluation comment*, 1(2):n2, 1968.

[2] M. M. Bronstein, J. Bruna, T. Cohen, and P. Velickovic. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *CoRR*, abs/2104.13478, 2021.

[3] Y. Chen, P.-H. Wuillemin, and J.-m. Labat. Discovering prerequisite structure of skills through probabilistic association rules mining. In *Proceedings of the 8th International Conference on Educational Data Mining*, pages 117–124, 06 2015.

[4] A. T. Corbett. Cognitive computer tutors: Solving the two-sigma problem. In *Proceedings of the 8th International Conference on User Modeling 2001*, page 137–147, 2001.

[5] A. Grover and J. Leskovec. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 855–864, 2016.

[6] A. Katharopoulos and F. Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2525–2534, 2018.

[7] A. Katharopoulos and F. Fleuret. Processing

megapixel images with deep attention-sampling models. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 3282–3291. PMLR, 2019.

[8] K. R. Koedinger, A. T. Corbett, and C. Perfetti. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cogn. Sci.*, 36:757–798, 2012.

[9] B. Kulis and M. I. Jordan. Revisiting k-means: New algorithms via bayesian nonparametrics. *ICML*, 2012.

[10] B. Kulis and M. I. Jordan. Revisiting k-means: New algorithms via bayesian nonparametrics. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[11] K. Leelawong and G. Biswas. Designing learning by teaching agents: The betty's brain system. *Int. J. Artif. Intell. Ed.*, 18(3):181–208, Aug. 2008.

[12] Y. Liu, Y. Yang, X. Chen, J. Shen, H. Zhang, and Y. Yu. Improving knowledge tracing via pre-training question embeddings. *arXiv preprint arXiv:2012.05031*, 2020.

[13] N. Maharjan, D. Gautam, and V. Rus. Assessing free student answers in tutorial dialogues using LSTM models. In *Artificial Intelligence in Education - 19th International Conference, AIED*, pages 193–198, 2018.

[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems*, pages 3111–3119, 2013.

[15] D. M. Morrison, B. Nye, V. Rus, S. Snyder, J. Boller, and K. B. Miller. Tutorial dialogue modes in a large corpus of online tutoring transcripts. In *Artificial Intelligence in Education - 17th International Conference*, volume 9112, pages 722–725. Springer, 2015.

[16] H. Nakagawa, Y. Iwasawa, and Y. Matsuo. Graph-based knowledge tracing: modeling student proficiency using graph neural network. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 156–163, 2019.

[17] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever. Deep double descent: Where bigger models and more data hurt. In *ICLR*, 2020.

[18] S. Pandey and G. Karypis. A self attentive model for knowledge tracing. In *EDM*. International Educational Data Mining Society (IEDMS), 2019.

[19] B. E. Penteado. Estimation of prerequisite skills model from large scale assessment data using semantic data mining. In *International Conference on Educational Data Mining*, pages 675–677, 2016.

[20] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

[21] S. Ritter. Communication, cooperation and competition among multiple tutor agents. In *Artificial Intelligence in Education: Knowledge and media in learning systems*, pages 31–38, 1997.

[22] S. Ritter, R. Baker, V. Rus, and G. Biswas. Identifying strategies in student problem solving. *Design Recommendations for Intelligent Tutoring Systems*, 7:59–70, 2019.

[23] S. Ritter, M. Yudelson, S. E. Fancsali, and S. R. Berman. How mastery learning works at scale. In *L@S*, pages 71–79. ACM, 2016.

[24] Y. Roh, K. Lee, S. E. Whang, and C. Suh. Fairbatch: Batch selection for model fairness. In *9th International Conference on Learning Representations, ICLR*, 2021.

[25] V. Rus, S. K. D'Mello, X. Hu, and A. C. Graesser. Recent advances in conversational intelligent tutoring systems. *AI Magazine*, 34(3):42–54, 2013.

[26] V. Rus, N. Maharjan, L. J. Tamang, M. Yudelson, S. R. Berman, S. E. Fancsali, and S. Ritter. An analysis of human tutors' actions in tutorial dialogues. In *Proceedings of the International Florida Artificial Intelligence Research Society Conference*, pages 122–127, 2017.

[27] A. Shakya, V. Rus, and D. Venugopal. Student strategy prediction using a neuro-symbolic approach. In *Proceedings of the 14th International Educational Data Mining Conference (EDM 21)*, 2021.

[28] T. Smith and M. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.

[29] X. Song, J. Li, Q. Lei, W. Zhao, Y. Chen, and A. Mian. Bi-clkt: Bi-graph contrastive learning based knowledge tracing. *Knowledge-Based Systems*, 241:108274, 2022.

[30] J. C. Stamper, K. R. Koedinger, R. S. J. de Baker, A. Skogsholm, B. Leber, S. Demi, S. Yu, and D. Spencer. Datashop: A data repository and analysis service for the learning science community. In *AIED*, volume 6738, page 628, 2011.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[32] D. Venugopal and V. Rus. Joint inference for mode identification in tutorial dialogues. In *COLING 2016, 26th International Conference on Computational Linguistics*, pages 2000–2011. ACL, 2016.

[33] D. Venugopal, V. Rus, and A. Shakya. Neuro-symbolic models: A scalable, explainable framework for strategy discovery from big edu-data. In T. W. Price and S. S. Pedro, editors, *Joint Proceedings of the Workshops at the International Conference on Educational Data Mining 2021 (EDM 2021)*, 2021.

[34] J. Wong, M. Khalil, M. Baars, B. D. Koning, and F. Paas. Exploring sequences of learner activities in relation to self-regulated learning in a massive open online course. *Computers & Education*, 2019.

[35] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi. Fairness constraints: A flexible approach for fair classification. *Journal of Machine Learning Research*, 20(75):1–42, 2019.

[36] W. Zhao, J. Xia, X. Jiang, and T. He. A novel framework for deep knowledge tracing via gating-controlled forgetting and learning mechanisms. *Information Processing and Management*, 60(1):103114, 2023.

Exploring the Effectiveness of Vocabulary Proficiency Diagnosis Using Linguistic Concept and Skill Modeling

Boxuan Ma
Kyushu University
OpenDNA Inc.
boxuan@artsci.kyushu-u.ac.jp

Gayan Prasad Hettiarachchi
OpenDNA Inc.
Tokyo, Japan
gayan@open-dna.jp

Sora Fukui
OpenDNA Inc.
Tokyo, Japan
fukui@open-dna.jp

Yuji Ando
OpenDNA Inc.
Tokyo, Japan
ando@open-dna.jp

ABSTRACT

Vocabulary proficiency diagnosis plays an important role in the field of language learning, which aims to identify the level of vocabulary knowledge of a learner through his or her learning process periodically, and can be used to provide personalized materials and feedback in language-learning applications. Traditional approaches are widely applied for modeling knowledge in science or mathematics, where skills or knowledge concepts are well-defined and easy to associate with each item. However, only a handful of works focus on defining knowledge concepts and skills using linguistic characteristics for language knowledge proficiency diagnosis. In addressing this, we propose a framework for vocabulary proficiency diagnosis based on neural networks. Specifically, we propose a series of methods based on our framework that uses different linguistic features to define skills and knowledge concepts in the context of the language learning task. Experimental results on a real-world second-language learning dataset demonstrate the effectiveness and interpretability of our framework. We also provide empirical evidence with ablation testing to prove that our knowledge concept and skill definitions are reasonable and critical to the performance of our model.

Keywords

Deep learning, Cognitive diagnosis, Vocabulary proficiency, Linguistic skill

1. INTRODUCTION

Vocabulary proficiency diagnosis is one of the key fundamental technologies supporting language education and has lately gained increased popularity in online language learning. It is crucial to identify the learners' latent proficiency level on different knowledge concepts (e.g., words) to higher accuracy in providing personalized materials and adaptive feedback in language-learning applications [1]. In practice, with the diagnostic results, systems can provide further support, such as learning planning, learning material recommendation, and computerized adaptive testing accordingly. Most importantly, it can help second-language learners to place themselves in the correct learning space or level after a long gap without using the application, during which they might have forgotten a lot or, conversely, have advanced in the target language without the use of the application [25].

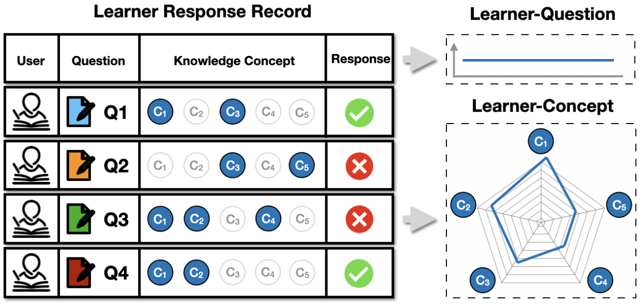


Figure 1: An example of cognitive diagnosis.

Many cognitive diagnosis methods have been proposed for knowledge proficiency diagnosis of learners. Figure 1 shows a simple example of a cognitive diagnosis system, which consists of learners, question items, knowledge concepts, and learner responses (scores). Specifically, a learner interacts with a set of questions and leaves their responses. Moreover, human experts usually label each question item with several knowledge concepts. Then, the goal is to infer their actual knowledge proficiency based on the interactions. Therefore, a cognitive diagnosis system can be abstracted as a learner-question-concept interaction modeling problem, and most previous works focus on learner-question interaction models or learner-concept interaction models [11]. For example, traditional methods like Item Response Theory (IRT) [9], Multidimensional IRT (MIRT) [24], and Matrix Factorization (MF) [23] try to model the learner-question interaction

B. Ma, G. P. Hettiarachchi, S. Fukui, and Y. Ando. Exploring the effectiveness of vocabulary proficiency diagnosis using linguistic concept and skill modeling. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 149–159, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115675

and provide learner latent traits (e.g., ability level) and the question features (e.g., difficulty level). In addition, MIRT and MF cannot provide explainable traits and IRT only provides an overall latent trait for learners, while each question usually assesses different knowledge concepts or skills. Other works such as Deterministic Inputs, Noisy-And gate (DINA) [6] try to build the learner-concept interaction instead of learner-question interaction. Unlike learner-question interaction models, learner-concept interaction models could infer the learner's traits in detail for each knowledge concept contained in the question item, despite leaving information of questions underexploited by simply replacing them with their corresponding concepts. Although great successes have been made, there are some limitations of traditional methods, which decay their effectiveness. Also, these approaches are widely applied for modeling knowledge in science or mathematics and ignore characteristics of language learning, which make it a significant research challenge to infer the mastery level of learners' vocabulary proficiency.

A critical drawback of traditional methods is that they can only exploit the response results and ignore the actual contents and formats of the items and cannot effectively utilize the rich information hidden within question texts and underlying formats [18]. Most traditional methods were proposed for scale-based tests, where a group of examinees is tested using the same small set of questions, and each examinee is supposed to respond to every question. As a result, the response data is complete and usually not large. While for learning applications nowadays, the data might be collected via different scenes, such as offline examinations and online self-regulated learning, and the distribution of response data can be of high volume but very sparse due to the large total number of items and limited questions attempted by the learners [33]. Therefore, neglecting contents and formats leaves traditional methods no possibility to utilize the relationships of different items, hence they are unable to generalize item parameters to unseen items [25]. Previous studies have already shown that the information of questions is significantly related to item parameters, for instance, the difficulty level. For language vocabulary questions, character length and corpus frequency prove to be essential factors for predicting vocabulary difficulty [5], while the average word and sentence lengths have been used as key features to predict text difficulty [2, 25]. Also, studies have indicated that different question formats impact the difficulty level and explanatory power in predicting receptive skills [16]. For the same vocabulary, different question formats are often used collectively to assess different skills, such as reading, writing, listening, and speaking skills, and many assessments have a mixture of item types. Consequently, it is important to consider the format information of the items and their influence on different traits when building a vocabulary proficiency diagnosis model.

Another important challenge is to define and use linguistic skills for vocabulary proficiency diagnosis. Although many approaches are widely applied for proficiency diagnosis, they have not frequently been applied to data generated in language learning settings. Instead, they have been primarily applied to science, engineering, and mathematics learning contexts, where skills or knowledge concepts are well-defined and easy to associate with each item. Most works use manually labeled Q-matrix to represent the knowledge relevancies of each question. For example, a math question: $6 \times 9 + 3 = ()$ examines the mastery of two knowledge concepts: Addition and Multiplication. Thus, the Q-matrix for this question could be labeled as $(1, 1, 0, ..., 0)$, where the first two positions show this question test Addition and Multiplication concepts, and other positions are labeled with zero, indicating other knowledge concepts are not included. However, proficiency diagnosis in the realm of language learning is different from other domains since linguistic skills are hard to define and need to be well-designed [21, 38].

To address these challenges, which have not been well explored in the research community, in this paper, we propose a framework for vocabulary proficiency diagnosis, which could capture the learner-question interactions more accurately using neural networks. In addition, we use linguistic features of words such as morphological and semantic features to define knowledge concepts and skills related to vocabulary and grammar knowledge that is shared between words. Extensive experimental results on a real-world second-language learning dataset demonstrate the effectiveness and interpretational power of our proposed framework. We also provide empirical evidence with ablation testing to prove that our knowledge concept and skill definitions are reasonable and critical to the performance of our model. The results show that using linguistic features to refine knowledge concepts and skills improves performance over the basic word-level model. We also explore the relationship of the question format, and in turn, its effect on the vocabulary proficiency diagnosis.

2. RELATED WORK

2.1 Cognitive Diagnosis

Cognitive diagnosis is a fundamental and important task, and many classical cognitive diagnosis models have been developed in educational psychology, such as IRT, MIRT, and DINA. IRT [9] is a widely used method and has been applied in educational testing environments since the 1950s [9]. It applies the logistic-like item response function and provides interpretable parameters. In its simplest form, IRT could be written as:

$$P(X_{ij} = 1) = \sigma(\theta_i - \beta_j),$$

where P is the probability of the learner i answering the item j correctly, σ is a logistic-like function, θ and β are unidimensional and continuous latent traits, indicating learner ability and item difficulty, respectively. Besides the basic IRT, other IRT models extend the basic one by factoring in other parameters, such as the item discrimination or guessing parameter.

IRT has proven to be a robust model. However, a single ability dimension is sometimes insufficient to capture the relevant variation in human responses. By extending the trait features into multidimensions, Reckase et al. [24] proposed MIRT, which tries to meet multidimensional data demands by including an individual's multidimensional latent abilities for each skill. MIRT goes a step further compared to IRT, however, as the process of estimating the parameters for MIRT is the same as IRT, these two models share the same shortcomings [4]. Also, latent trait vectors provided by

IRT and MIRT is not explainable enough to guide learners' self-assessment [34].

By characterizing learner features (e.g., ability) and item features (e.g., difficulty), IRT builds learner-question interaction and provides an overall latent trait for learners. However, real-world questions usually assess different knowledge concepts or skills, and an overall trait result is insufficient [20]. To provide detailed results on each knowledge concept or skill, other works try to directly build learner-concept interaction. For example, DINA [6] model the learner-concept interaction by mapping questions to corresponding concepts/ skills directly with Q-matrix, which indicates whether the knowledge concept is required to solve the question. Different from IRT, θ and β are multi-dimensional and binary in DINA, where β came directly from Q-matrix. Another two parameters, guessing g and slipping s, are also taken into consideration. The DINA formula is written as:

$$P(X_{ij} = 1) = {g_j}^{1-\eta_{ij}}(1-s_j)^{\eta_{ij}}, \;\; \eta_{ij} = \prod_{k=1}^{K} {\theta_{ik}}^{\beta_{jk}},$$

where the latent response variable η_{ij} indicates whether the learner has mastered all the required knowledge to solve the question. And the probability of the learner i correctly answering item j is modeled as the compound probability that the learner has mastered all the skills required by the question without slip, and the learner does not master all the required skills but makes a successful guess. Although DINA has made great progress and shows its advantage compared to IRT in specific scenarios, it ignores the features of questions and simply replaces them with the corresponding knowledge concepts/skills, thus leaving useful information from questions underexploited.

2.2 Matrix Factorization

Besides the traditional models, the other line of studies has demonstrated the effectiveness of MF for predicting learner performance by factorizing the score matrix, which was originally widely used in the field of recommendation systems [3]. Studies have shown that predicting learner performance can be treated as a rating prediction problem since *learner*, *question*, and *response* can correspond to *user*, *item*, and *rating* in recommendation systems, respectively.

Toscher et al. [30] applied several recommendation techniques in the educational context, such as Collaborative Filtering (CF) and MF, and compared them with traditional regression methods for predicting learner performance. Along this line, ThaiNghe et al. [28] proposed multi-relational factorization models to exploit multiple data relationships to improve the prediction results in intelligent tutoring systems. In addition, Desmarais [8] used Non-negative Matrix Factorization (NMF) to map question items to skills, and the resulting factorization allows a straightforward interpretation in terms of a Q-matrix. Similarly, Sun et al. [27] proposed a method that uses Boolean Matrix Factorization (BMF) to map items into latent skills based on learners' responses. Wang et al. [36] proposed a Variational Inference Factor Analysis framework (VarFA) and utilized variational inference to estimate learners' mastery level of each knowledge concept. Despite their effectiveness in predicting learner performance, the latent trait vectors in MF are not interpretable for cognitive diagnosis, i.e., there is no clear correspondence between elements in trait vectors and specific knowledge concepts. Also, these works have considered only learners and question items, and ignored other information that may also be useful.

2.3 Deep-learning based models

With the recent surge in interest in deep learning, many works have begun to use deep learning to address some of the shortcomings of traditional cognitive diagnosis models [13, 19, 29].

Traditional methods are often based on simple linear functions, such as the logistic-like function in IRT or the inner product in matrix factorization, which may not be sufficient. To improve precision and interpretability, some previous works focus on interaction function design and use neural networks to learn more complex non-linear functions. For example, Wang et al. [33] propose a Neural Cognitive Diagnosis (NCD) framework for Intelligent Education Systems, which leverages neural networks to automatically learn the interaction function.

Some researchers focus on incorporating the content representation from question texts into the model by neural networks, which is difficult with traditional methods. Cheng and Liu [4] proposed a general Deep Item Response Theory (DIRT) framework that uses deep learning to estimate item discrimination and difficulty parameters by extracting information from item texts. Wang et al. [34] applied neural networks to extract two typical types of information in the question text: knowledge concepts and extra text-related factors. Their results indicated that using such content information benefited the model and significantly improved its performance.

Other deep-learning models try to incorporate dependency relations among knowledge concepts for enhancing diagnosis performance. For example, Wang et al. [35] proposed a model based on neural networks and aggregate knowledge relationships by converting all knowledge concepts into a graph structure. Ma et al. [22] proposed the Prerequisite Attention model for Knowledge Proficiency (PAKP) to explore the prerequisite relation among knowledge concepts and use it for inferring knowledge proficiency. Recent work proposed the Relation map driven Cognitive Diagnosis (RCD) [11] model by comprehensively modeling the learner-question interactions and question-concept relations. Their model achieved better performance compared to traditional works that consider only learner-question interactions (e.g., IRT) or only question-concept interactions (e.g., DINA).

Although deep learning models have been widely explored nowadays, they have been primarily applied to learning contexts such as math, algebra, or science, where skills or knowledge concepts are well-defined and easily associated with each item. Therefore, these methods cannot be directly used in the language learning area, and linguistic skills need to be well-defined and well-designed for language proficiency diagnosis. In addition, except for the work by wang et al. [34], other aforementioned works failed to consider question for-

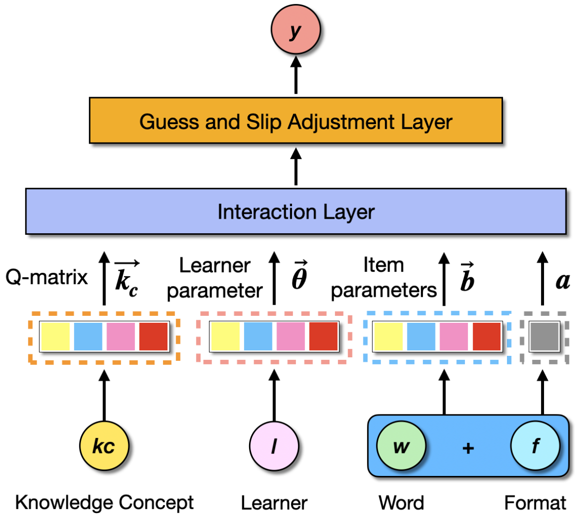


Figure 2: Overview of the proposed framework.

mats, which are important for language-learning questions and may have a significant influence on the question difficulty level and learner's performance.

3. PROPOSED METHOD

We first give the definition of our problem in Section 3.1. Then we present our proposed framework in Section 3.2.

3.1 Problem Formulation

Like every test, there are two basic elements: *user* and *item*, where a user represents a learner, and an item represents a question. We use L to denote a set of learners, Q to denote a set of questions and s to denote the learner-question interaction score. Learner question records are represented by $R = \{(l, q, s) | l \in L, q \in Q, s \in \{0, 1\}\}$, which means learner l responded to question q and received the score s. Each score s is in $\{0, 1\}$ where 1 indicates the question is correctly answered while 0 stands in the opposite.

Given enough question-records data R of learners, our goal is to build a model to mine learners' proficiency through the task of performance prediction.

3.2 Framework

Generally, for a cognitive diagnostic system, there are three parts that need to be considered: learner, question item, and interaction function. As shown in Figure 2, we propose a cognitive diagnostic framework with deep learning, which aims to obtain the learner parameter (proficiency) and item parameters (discrimination and difficulty). Specifically, for each response log, we use one-hot vectors of the corresponding learner and question as input and obtain the diagnostic parameters of the learner and question. Then the model learns the interaction function among the learner and item parameters and outputs the probability of correctly answering the question. After training, we get the learner's proficiency vectors as diagnostic results.

3.2.1 *Item Parameters*

The item's characteristics are calculated in the item network to represent the traits of a specific item. Two parameters extended from the Two-Parameter Logistic IRT model [32] are used in our model, i.e., discrimination and difficulty. The discrimination $a \in (0, 1)$ indicates the ability of an item to differentiate among learners whose knowledge mastery is high from those with low knowledge mastery, and difficulty $\boldsymbol{b} \in (0, 1)^{1 \times K}$ indicates the difficulty of each knowledge concept examined by the question, where K is the number of knowledge concepts.

As we mentioned before, two elements influence the item's characteristics for a vocabulary question: the target word and the specific item format. Then the item is represented by integrating the one-hot word embedding vector $\boldsymbol{w}$ and one-hot item format embedding $\boldsymbol{f}$.

$$\boldsymbol{i} = \boldsymbol{w} \oplus \boldsymbol{f}, \tag{1}$$

where $\oplus$ is the concatenation operation. After obtaining item representation using the word embedding and item format, we input it into two different networks to estimate the question discrimination a and knowledge difficulty $\boldsymbol{b}$. Specifically:

$$a = \sigma(F_a(\boldsymbol{i})), \tag{2}$$

$$\boldsymbol{b} = \sigma(F_b(\boldsymbol{i})) \tag{3}$$

Where F_a and F_b are discrimination and difficulty networks, respectively, and σ is the sigmoid function.

3.2.2 *Learner Parameter*

In the learner network, the proposed method characterizes the traits of learners, which is closely related to the proficiency of various knowledge concepts or skills tested in the question and would affect the learner's performance. Specifically, each learner is represented with a proficiency vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)$, where $\theta_i \in [0, 1]$ represents the degree of proficiency of a learner on a specific knowledge concept or skill i and the goal of our cognitive diagnosis model is to mine learners' proficiency through the task of performance prediction. The proficiency vector is obtained by multiplying the learner's one-hot representation vector l with a trainable matrix $\boldsymbol{A}$. That is:

$$\boldsymbol{\theta} = \boldsymbol{l} \times \boldsymbol{A}. \tag{4}$$

3.2.3 *Prediction of Learner Response*

Interaction layer. The proposed method predicts a learner's response performance to a question as a probability. We input the representations of the learner parameter and question parameters (i.e., item discrimination and knowledge difficulty, respectively) into an interaction function to predict the learner's probability of answering the specific question correctly.

The interaction function simulates how learner parameters interact with question parameters to get the response results, for example, a simple logistic-like function is used as the interaction function in IRT. Based on previous works [22, 33, 34, 35], we use a neural network to learn a more complex non-linear interaction function to boost the model.

Specifically, the input of the interaction function can be formulated as:

$$\boldsymbol{x} = a\ (\boldsymbol{\theta} - \boldsymbol{b}) \odot \boldsymbol{k_c} \tag{5}$$

where $\boldsymbol{k_c}$ is the knowledge concept or skill vector that indicates the relationship between the question and knowledge concepts or skills, which is usually pre-labeled by experts and obtained directly from Q-matrix. We discuss how we define the knowledge concepts or skills in Section 3.3. The operator $\odot$ is the element-wise product and $\boldsymbol{x}$ indicates the learner's performance on each concept pertaining to the question. We then use a three-layer feed-forward neural network F_i to learn the non-linear activation function and output the probability p that the learner answers the question correctly. It can be formulated as:

$$p = \sigma(F_i(\boldsymbol{x})). \tag{6}$$

Following previous works [33, 34, 35], we restrict each weight of F_i to be positive during the process of training to ensure the monotonicity assumption, which assumes that the probability of learners answering the exercise correctly increases monotonically with the degree of mastery on each knowledge concept pertaining to the question.

Guess and Slip Adjustment. We noticed that many question items in the dataset are multiple-choice items, which makes it highly possible for the learners to guess the correct answer even if they don't master the knowledge concept, or slip even though they know the answer. To obtain better results, we add a guessing parameter $g \in [0,\ 1]$ and a slipping parameter $s \in [0,\ 1]$ to adjust the performance results, where g indicates the probability that a learner did not master the knowledge concepts but guessed the correct answer and s indicates the probability that a learner masters the knowledge concepts but did not answer correctly. The guessing and slipping parameters can be formulated as:

$$g = \sigma(F_g(\boldsymbol{i} \oplus \boldsymbol{l})), \tag{7}$$

$$s = \sigma(F_s(\boldsymbol{i} \oplus \boldsymbol{l})), \tag{8}$$

where F_g is the guessing and F_s is the slipping networks, respectively. To compute the final probability that a learner answers the question correctly, we apply adjustments of the guessing parameter and slipping parameter on the probability estimation, which can be expressed as:

$$y = g + (s - g) \times p. \tag{9}$$

3.2.4 Model Learning

We use the binary cross-entropy loss function for the proposed method. The learner's score is recorded as 1 when she/he answers the item correctly and 0 otherwise. For learner i and question j, let y_{ij} be the actual score for learner i on question j, and $\hat{y}_{ij}$ be the predicted score. Thus, the loss for learner i on question j is defined as:

$$\mathcal{L} = y_{ij} log\hat{y}_{ij} + (1 - y_{ij})log(1 - \hat{y}_{ij}). \tag{10}$$

Using Adam optimization [15], all parameters are learned simultaneously by directly minimizing the objective function. After training, the value of $\boldsymbol{\theta}$ is what we get as the diagnostic result, which denotes the learner's knowledge proficiency.

Table 1: An example subwords Q-matrix.

	Knowledge Concept									
Words	active	actual	actor	act	-tive	-tual	-ual	-tor	-or	⋯
active	1	0	0	1	1	0	0	0	0	⋯
actual	0	1	0	1	0	1	1	0	0	⋯
actor	0	0	1	1	0	0	0	1	1	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

3.3 Defining Knowledge Concepts and Skills

The knowledge concept or skill vector indicates the relationship between question items and knowledge concepts/skills, which is fundamentally essential as we need to diagnose the degree of proficiency of a learner corresponding to a specific knowledge concept/skill. As for each question, the knowledge concept/skill vector $\boldsymbol{c} = (c_1, c_2, c_3, \dots c_k), c_i \in \{0, 1\}$ represents if a specific knowledge concept/skill is required to solve the question, in which $c_i = 1$ indicates that the knowledge concept/skill is included in the question and conversely $c_i = 0$ is not.

Usually, skills or knowledge concepts are pre-labeled by experts, and the vector **c** can be directly obtained from the pre-given Q-matrix. However, the knowledge concept/skill is difficult to define for language learning compared to other learning contexts such as science, engineering, and mathematics. Conventional models treat all question items nested under a particular word equivalent, but even for the same word, the ability of learners to comprehend a specific word can be divided into different levels. Some researchers define 'word knowledge' as different components including spelling, word parts, meaning, grammatical functions, the associations a word has with other words, and collocation to describe the totality of the learner's knowledge of a specific word in a language [20]. Thus, different items may refer to the same word if the word is used differently in multiple contexts (e.g., used as different parts of speech), or if different components of the word are tested. It is important to consider these when building vocabulary proficiency diagnosis models.

In the following subsections, we introduce several methods for defining knowledge concepts/skills in vocabulary proficiency diagnosis using different linguistic features and provide more detailed results on diagnosing associated knowledge concepts/skills.

3.3.1 Words as Knowledge Concepts

The simplest way to label knowledge concepts in an item is to simply use the unique words as knowledge concepts. There could be many knowledge concepts (e.g., many unique words) in a language-learning system, but only one knowledge concept (i.e., a word tested in the question) is related to a question item.

3.3.2 Sub-words as Knowledge Concepts

Another way to label multiple knowledge concepts in an item is to identify sub-words that comprise a word and treat each of these sub-words as an additional knowledge concept. Sub-words can be viewed as morphological features of an original word, which may indicate the relationships of different

Table 2: Summary of question formats and required skill(s).

Format	Skill	Q-matrix Vector
F1	Recognition	[1, 0, 0, 0]
F2	Recognition	[1, 0, 0, 0]
F3	Recognition, Listening	[1, 1, 0, 0]
F4	Recognition, Spelling	[1, 0, 1, 0]
F5	Reading	[0, 0, 0, 1]

words and reinforce the knowledge related to gender agreement, prefixes, suffixes, compound words, etc. Inspired by the work of Zylich and Lan [38], we apply a sub-word tokenizer to automatically identify sub-words contained in each word. As shown in Table 1, we formulate a Q-matrix to apply the sub-word knowledge concepts for each word. For example, the word 'active' could have additional knowledge concepts such as 'act' and '-tive'.

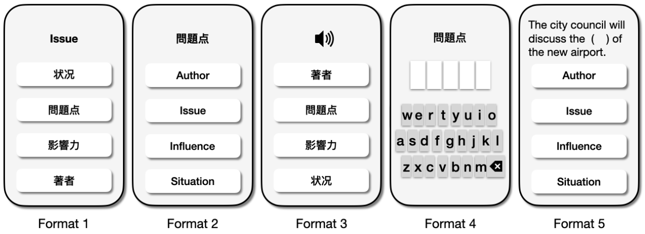


Figure 3: Examples of different question formats

3.3.3 *Semantically Similar Words as Knowledge Concepts*

Recent works indicated that cross-effects commonly exist in language learning [21, 38]. That is, during the exercise process of a learner, when an exercise of a particular knowledge concept is given, she/he also applies the relevant knowledge concepts to solve it. Specifically, in language learning, it seems that knowledge pertaining to semantically-similar words related to the word being tested are helpful in answering the question.

Following previous work [38], we used word embeddings to obtain semantic similarities of words. First, we embedded each word into a 300-dimensional vector using pre-trained fastText word embeddings [12] and calculated the cosine similarity scores between each pair of words to get a matrix of values that indicates the similarities of each word. Using this similarity matrix, all the similar words in the dataset that have cosine similarity larger than a threshold α with the current word can be counted as addition knowledge concepts required to solve the question. The threshold α is used to control the degree of semantic similarity, for example, only highly semantically similar words can be used as knowledge concepts in the Q-matrix if α is large, and if $\alpha = 1$, this model reduces back to the basic word-level model that only uses the current word as the knowledge concept. Otherwise, if $\alpha = 0$, which means that all other words that have non-negative similarity with the current word are treated as knowledge concepts.

3.3.4 *High-order Skills*

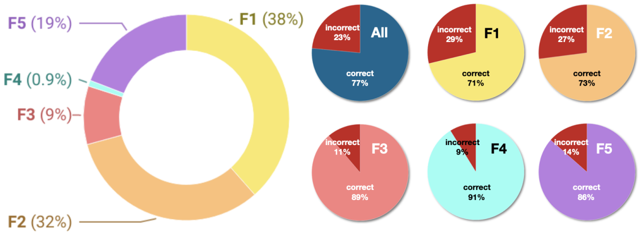


Figure 4: Distribution of question formats and response pie chart.

We formulated several methods for defining knowledge concepts in language proficiency diagnosis using different linguistic features such as additional morphological and semantic concepts. However, the ability used to solve vocabulary questions can depend on several high-order skills but not on whether the learner knows the word or not. Following previous works [14, 20, 37], we also consider defining skills instead of knowledge concepts in language proficiency diagnosis.

Here we propose two different methods to label skills in language proficiency. The most basic way we can choose to label a skill is by the question format. As shown in Figure 3, there are five different question formats in our dataset (more detailed information on the data can be found in Section 4.1). And if a learner is good at correctly answering a particular type of question, we can assume that she/he has a high skill in this question format. However, there will only be a single skill associated with each item and is not explainable enough if we use the question format as skills. To have a better interpretation, as summarized in Table 2, for each question format (see Figure 3), we defined some high-order language skills (i.e., Recognition, Listening, Spelling, Reading) required to tackle a specific question format based on some of the evidence from the literature [14, 16, 20, 26].

4. EVALUATION

4.1 Dataset

Our real-world dataset came from one of Japan's most popular English-language learning applications, and most of the users are Japanese students. The dataset includes 9,969,991 learner-item interactions from 2,014 users. There are 1,900 English words in the dataset, and each word has five different question formats collectively assessing different skills, resulting in 9500 items. The different question formats are shown in Figure 3, and some basic statistics of the dataset and response distributions are shown in Figure 4.

4.2 Experimental Settings

4.2.1 *Evaluation Metrics*

The performance of a cognitive diagnosis model is hard to evaluate as we can't obtain the true knowledge proficiency of learners directly. Usually, the models are evaluated by predicting learner performance in most cognitive diagnosis works. Following previous works, we evaluated by comparing the predicted responses with the ground truth, i.e., the actual response by the learners.

To set up the experiment, the data were randomly split into

80%/20% for training and test purposes, respectively. We filtered out the learners who had answered less than 50 questions so that every learner could be diagnosed with enough data. Like previous works [4, 34, 35], we use Prediction Accuracy (ACC), Area Under Curve (AUC), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) as metrics. The larger the values of ACC and AUC, and the smaller the values of MAE and RMSE, the better the results are.

4.2.2 Comparison

We name our model as Vocabulary Proficiency Diagnosis Model (VPDM) and compared our models using different knowledge concept and skill definitions with several existing models given below.

- DINA [6]: DINA is a cognitive diagnosis method that models learner concept proficiency by a binary vector.

- IRT [9]: IRT is a classical baseline method that models learners' and questions' parameters using the item response function.

- MIRT [24]: Extending from IRT, MIRT can model the multidimensional latent abilities of a learner.

- PMF [10]: Probabilistic matrix factorization (PMF) is a factorization method that can map learners and questions into the same latent factor space.

- NMF [17]: Non-negative matrix factorization (NMF) is also a factorization method, but it is non-negative, which can work as a topic model.

- NCD [33]: NCD is a recently proposed method that uses neural networks to learn more complex non-linear learner-question interaction functions.

Among these baselines, IRT, MIRT, and DINA are widely used methods in educational psychology. PMF and NMF are two matrix factorization methods from the recommendation system and data mining fields. NCD is a recently proposed model based on deep learning.

4.2.3 Parameter Settings

We implemented our model and other baselines in PyTorch. The model was trained with a batch size of 256. We used Adam optimizer with a learning rate of 0.001. The dropout rate is set to 0.2, and early stopping is applied to reduce overfitting.

5. RESULTS

5.1 Performance Prediction

The overall results on all four metrics are shown in Table 3 for all baseline methods and our models predicting learners' performance. VPDM-Word, VPDM-Subword, VPDM-Semantic, VPDM-FormatSkill, and VPDM-LangSkill are our models using words, subwords, semantically similar words, question formats, and language skills as knowledge concepts /skills, respectively. We observe that our models perform better than all other models, indicating the effectiveness of our framework. Among other baseline models, we noticed that the performance of NCD is comparable to our models and better than educational psychology methods (i.e., DINA, IRT, and MIRT) and matrix factorization methods (i.e., NMF and PMF), which demonstrates that leveraging deep learning could model the learner-question interactions more accurately than other conventional models.

Table 3: Performance comparison.

Model	ACC ↑	AUC ↑	MAE ↓	RMSE ↓
DINA	0.756	0.704	0.348	0.446
IRT	0.770	0.721	0.317	0.400
MIRT	0.768	0.728	0.311	0.399
NMF	0.768	0.722	0.355	0.405
PMF	0.771	0.731	0.328	0.398
NCD	0.772	0.734	0.316	0.397
VPDM-Word	0.773	0.736	0.309	0.396
VPDM-Subword	0.772	0.736	0.310	0.396
VPDM-Semantic	0.773	0.736	0.308	0.396
VPDM-FormatSkill	0.773	0.742	0.309	0.395
VPDM-LangSkill	0.773	0.742	0.308	0.395

In comparing our models, the performance of the VPDM-Word, VPDM-Subword, and VPDM-Semantic models are comparable, while VPDM-LangSkill and VPDM-FormatSkill models obtain better performance than other models, indicating that more broadly defined skills/knowledge concepts of an item are better. We will introduce our investigations to gain a deeper understanding of the difference among our models in the following subsections.

5.2 Impact of Different Formats

Many assessments have a mixture of item types (same as our dataset) since results based on a single format only reflect the knowledge unique to the specific format and might be misleading. To illustrate the performance of our models on different item formats, we separated the mixed-format dataset into different parts that only include different specific item formats, so we could conduct experiments to evaluate questions with a specific format. The results are shown in Figure 5 and the number of responses completed per learner is shown in Figure 6. Note that we did not test VPDM-LangSkill and VPDM-FormatSkill models here as they are intended for the mixed-format dataset.

Overall, the results indicate that our model consistently outperforms all other models. Furthermore, we observe that the prediction performance is affected by the question format, which highlights the fact that different question formats assess different traits.

5.3 Ablation Study

To investigate how the guessing and slipping adjustment layer affects model performance, we conducted some ablation experiments to compare the results. Table 4 shows the comparison results of the experiments on our mixed-format dataset and different single-format datasets. We observed that the performance improves when using the guessing parameter, and the model with guessing and slipping parameters obtained the best performance. It is reasonable as many items are multiple-choice in our dataset. In addition, we

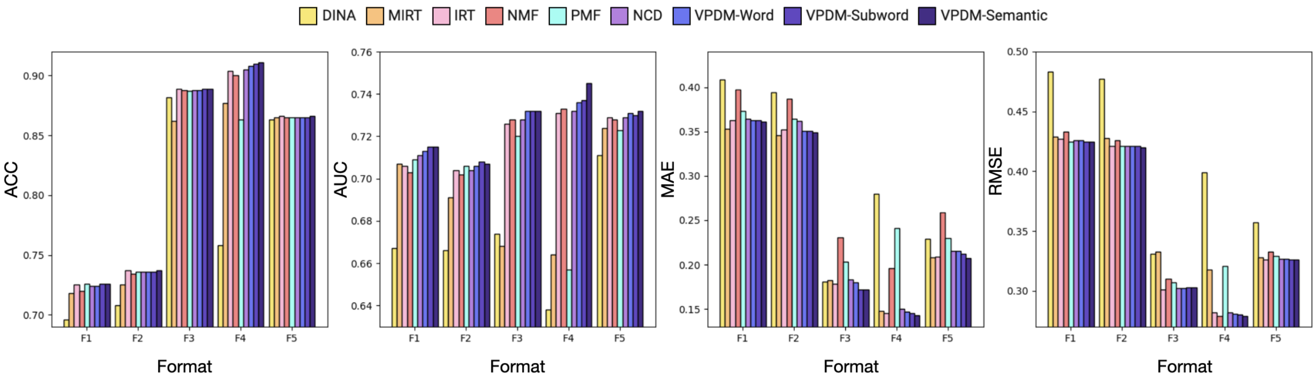
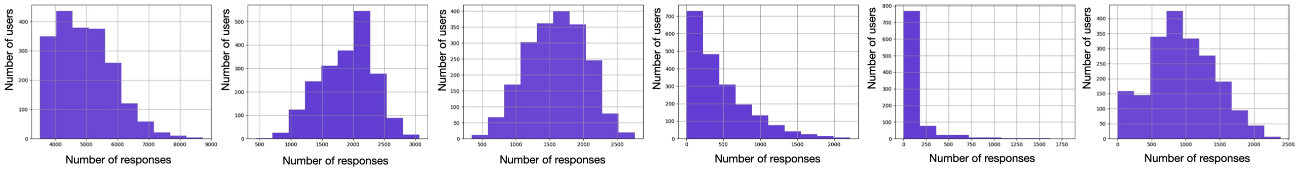


Figure 5: Comparison among different question formats.



(a) All items
(b) Format 1
(c) Format 2
(d) Format 3
(e) Format 4
(f) Format 5

Figure 6: Distribution of the number of responses per learner.

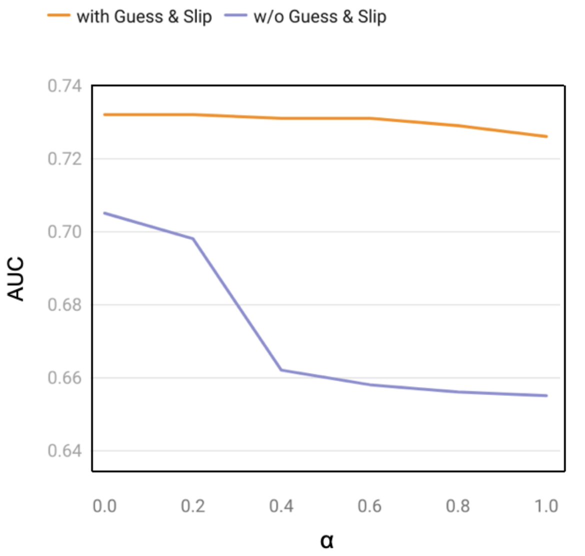


Figure 7: Comparative performance of semantically similar words as knowledge concepts via cosine similarity.

noticed that adding the slip and guessing parameters substantially improves some models' performance. This might imply that the Q-matrix is not specified appropriately in those models, though no formal rules exist to test this assumption [7].

In the comparison of the models that remove the guessing and slipping adjustment layer, the performance of the basic VPDM-Word model is the worst. As we expected, the knowledge assessed by a word item is not just simply related to the tested target word in the question. Moreover, the results confirm that the item's format carries meaning and is related to different traits, even though the questions with different formats are all designed for the same word.

As for subword and semantic models which use additional morphological or semantical knowledge concepts along with the tested target word, we observed improvements compared to the basic word-level model. One possible explanation is that the use of additional morphological or semantical knowledge concepts results in more items that share skills with each other, enabling the model to capture more interactions between learners and different words and reinforce the knowledge related to gender agreement, prefixes, suffixes, compound words, etc. [38]. For example, a closer inspection of the items revealed that even learners who are familiar with the word 'break' but do not know 'breakthrough' still have a good chance of answering some 'breakthrough' related items correctly. Figure 7 shows that varying the threshold parameter α in the VPDM-Semantic model does not influence the performance drastically. However, when we remove the guess and slip adjustment layer, we found that the performance of the model increases with the decreases of α, and the model performs best when $\alpha = 0$, which means that all other words that have non-negative similarity with the current word are treated as knowledge concepts. This result is in agreement with previous works, that an item designed to measure one trait may also require some level of other traits [37], and the proficiency of similar knowledge concepts can affect each other [11]. Specifically for language learning settings, it is important to focus not only on the interactions with the same word but also on interactions with other semantically similar words when predicting the degree of mastery of the target word [21]. We also noticed an intriguing finding for format 4, where VPDM-Subword and VPDM-Semantic outperformed the VPDM-Word model

Table 4: Results of the ablation study.

Model	Adjustment	All	F1	F2	F3	F4	F5
		AUC ↑	AUC ↑	AUC ↑	AUC ↑	AUC ↑	AUC ↑
VPDM-Word	-	0.655	0.668	0.669	0.685	0.628	0.715
	Guess	0.735	0.711	0.705	0.731	0.736	0.730
	Guess & Slip	0.736	0.713	0.706	0.732	0.736	0.731
VPDM-Subword	-	0.661	0.683	0.679	0.703	0.698	0.716
	Guess	0.734	0.711	0.703	0.729	0.731	0.729
	Guess & Slip	0.736	0.715	0.708	0.732	0.737	0.730
VPDM-Semantic	-	0.705	0.674	0.672	0.699	0.699	0.711
	Guess	0.734	0.713	0.706	0.730	0.732	0.731
	Guess & Slip	0.736	0.715	0.707	0.732	0.745	0.732
VPDM-FormatSkill	-	0.733	-	-	-	-	-
	Guess	0.740					
	Guess & Slip	0.742					
VPDM-LangSkill	-	0.735	-	-	-	-	-
	Guess	0.741					
	Guess & Slip	0.742					

significantly after the guess and slip adjustment layer was removed. This finding is particularly noteworthy because format 4 requires learners to type the word, and the results are more likely to be influenced by related morphological and semantic knowledge concepts such as prefixes, suffixes, and compound words. This result highlights the critical role of the item's format and how it influences the required knowledge in the question. Understanding this relationship between item format and knowledge requirements could potentially inform the design of more effective and efficient language learning assessments and improve learners' overall performance.

Finally, VPDM-LangSkill and VPDM-FormatSkill models obtain better performance than other models, indicating that more broadly defined skills and knowledge of an item are better in this task. For VPDM-FormatSkill model, one prevalent hypothesis is that items with different formats measure different traits or dimensions, and factors could be hypothesized to form on the basis of item format [31]. That is, the item's format might also be important and related to different traits or dimensions as suggested by previous works [7]. For VPDM-LangSkill model, the results show that learners' knowledge acquisition is influenced by high-order features (language abilities in this case). It greatly reduces the complexity of the model in cases where it is reasonable to view the examination as measuring several general abilities in addition to the specific knowledge states.

5.4 Interpretation of the Diagnosis

We visualize the diagnostic reports and evaluate the interpretation of the VPDM-LangSkill model as it is the most practical one with good performance. This visualization helps learners recognize their knowledge state intuitively and assists test developers to design question items effectively. As shown in Figure 8, we randomly sampled a learner and depict the proficiency diagnosed by IRT and VPDM-LangSkill. Each point on the radar diagram represents the mastery level of a certain trait. The red and blue lines denote the proficiency diagnosed by IRT and VPDM-LangSkill (scaled to $(0, 1)$), respectively. From the results, we can see that IRT only provides an overall unidimensional latent trait, the proficiency for all concepts is identical, therefore, it is not explainable enough to guide learners' self-assessment. As for the VPDM-LangSkill model, it is able to provide better interpretable insight for multidimensional traits (i.e., in our case, recognition, listening, spelling, and reading).

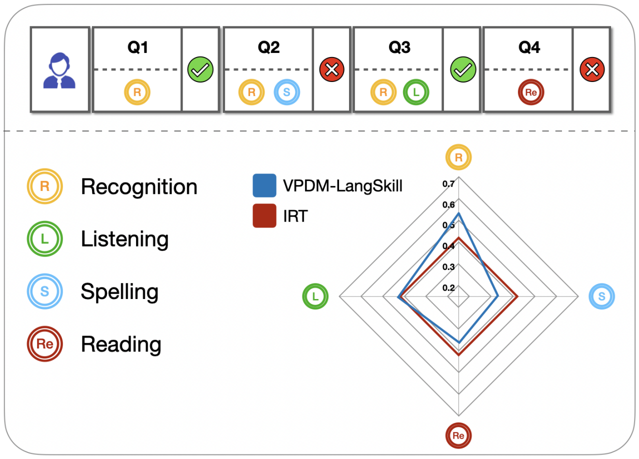


Figure 8: Visualization of a sample diagnostic report.

6. CONCLUSION

In this work, we proposed a framework for vocabulary proficiency diagnosis, which could capture the learner-question interactions more accurately using neural networks. In addition, we proposed a series of methods based on our framework, that uses different linguistic features to define skills and knowledge concepts in the context of a language learning task. Experimental results of cognitive diagnosis on real-world second-language learning dataset showed that the proposed approach outperforms existing approaches with higher accuracy and increased interpretability. We also provided empirical evidence with ablation testing to prove that our knowledge concept and skill definitions are reasonable and critical to the performance of our model.

There are some limitations in this work. Firstly, the learner base of the dataset is limited to learners of the same language background and thus might decrease the generalize of this work. We plan to test other datasets in future work. In addition, we only consider the target word that is tested in the question, however, some questions are multiple-choice, and some questions test contextual usage as the learner needs to fill in a sentence with the correct target word. Therefore, additional features such as context information and distractors in the question should also be considered as they also influence the learner's performance. We expect that this work will provide useful implications for language-learning applications that focus on vocabulary learning, and we will test more question formats and include additional linguistic skills to expand the capabilities of our model in future work.

7. ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number JP20H00622.

8. REFERENCES

[1] D. Avdiu, V. Bui, and K. P. Klimčíková. Predicting learner knowledge of individual words using machine learning. In *Proceedings of the 8th Workshop on NLP for Computer Assisted Language Learning*, pages 1–9, 2019.

[2] L. Beinborn, T. Zesch, and I. Gurevych. Predicting the difficulty of language proficiency tests. *Transactions of the Association for Computational Linguistics*, 2:517–530, 2014.

[3] Y. Chen, Q. Liu, Z. Huang, L. Wu, E. Chen, R. Wu, Y. Su, and G. Hu. Tracking knowledge proficiency of students with educational priors. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 989–998, 2017.

[4] S. Cheng, Q. Liu, E. Chen, Z. Huang, Z. Huang, Y. Chen, H. Ma, and G. Hu. Dirt: Deep learning enhanced item response theory for cognitive diagnosis. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2397–2400, 2019.

[5] B. Culligan. A comparison of three test formats to assess word difficulty. *Language Testing*, 32(4):503–520, 2015.

[6] J. De La Torre. Dina model and parameter estimation: A didactic. *Journal of educational and behavioral statistics*, 34(1):115–130, 2009.

[7] J. De La Torre and J. A. Douglas. Higher-order latent trait models for cognitive diagnosis. *Psychometrika*, 69(3):333–353, 2004.

[8] M. C. Desmarais. Mapping question items to skills with non-negative matrix factorization. *ACM SIGKDD Explorations Newsletter*, 13(2):30–36, 2012.

[9] S. E. Embretson and S. P. Reise. *Item response theory*. Psychology Press, 2013.

[10] N. Fusi, R. Sheth, and M. Elibol. Probabilistic matrix factorization for automated machine learning. *Advances in neural information processing systems*, 31, 2018.

[11] W. Gao, Q. Liu, Z. Huang, Y. Yin, H. Bi, M.-C. Wang, J. Ma, S. Wang, and Y. Su. Rcd: Relation map driven cognitive diagnosis for intelligent education systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 501–510, 2021.

[12] É. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[13] Z. Huang, Q. Liu, Y. Chen, L. Wu, K. Xiao, E. Chen, H. Ma, and G. Hu. Learning or forgetting? a dynamic approach for tracking the knowledge proficiency of students. *ACM Transactions on Information Systems (TOIS)*, 38(2):1–33, 2020.

[14] F. Kilickaya et al. Assessing l2 vocabulary through multiple-choice, matching, gap-fill, and word formation items. *Lublin Studies in Modern Languages and Literature*, 43(3):155–166, 2019.

[15] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[16] B. Kremmel and N. Schmitt. Interpreting vocabulary test scores: What do various item formats tell us about learners' ability to employ words? *Language Assessment Quarterly*, 13(4):377–392, 2016.

[17] D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13, 2000.

[18] Q. Liu, Z. Huang, Y. Yin, E. Chen, H. Xiong, Y. Su, and G. Hu. Ekt: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering*, 33(1):100–115, 2019.

[19] Q. Liu, R. Wu, E. Chen, G. Xu, Y. Su, Z. Chen, and G. Hu. Fuzzy cognitive diagnosis for modelling examinee performance. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(4):1–26, 2018.

[20] B. Ma, G. P. Hettiarachchi, and Y. Ando. Format-aware item response theory for predicting vocabulary proficiency. In *Proceedings of the 15th International Conference on Educational Data Mining*, pages 695–700, 2022.

[21] B. Ma, G. P. Hettiarachchi, S. Fukui, and Y. Ando. Each encounter counts: Modeling language learning and forgetting. In *LAK23: 13th International Learning Analytics and Knowledge Conference*, pages 79–88, 2023.

[22] H. Ma, J. Zhu, S. Yang, Q. Liu, H. Zhang, X. Zhang, Y. Cao, and X. Zhao. A prerequisite attention model for knowledge proficiency diagnosis of students. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4304–4308, 2022.

[23] A. Mnih and R. R. Salakhutdinov. Probabilistic matrix factorization. *Advances in neural information*

processing systems, 20, 2007.

[24] M. D. Reckase. Multidimensional item response theory models. In *Multidimensional item response theory*, pages 79–112. Springer, 2009.

[25] F. Robertson. Word discriminations for vocabulary inventory prediction. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1188–1195, 2021.

[26] L. S. Stæhr. Vocabulary size and the skills of listening, reading and writing. *Language Learning Journal*, 36(2):139–152, 2008.

[27] Y. Sun, S. Ye, S. Inoue, and Y. Sun. Alternating recursive method for q-matrix learning. In *Educational data mining 2014*, 2014.

[28] N. Thai-Nghe and L. Schmidt-Thieme. Multi-relational factorization models for student modeling in intelligent tutoring systems. In *2015 Seventh international conference on knowledge and systems engineering (KSE)*, pages 61–66. IEEE, 2015.

[29] S. Tong, Q. Liu, R. Yu, W. Huang, Z. Huang, Z. A. Pardos, and W. Jiang. Item response ranking for cognitive diagnosis.

[30] A. Toscher and M. Jahrer. Collaborative filtering applied to educational data mining. *KDD cup*, 2010.

[31] R. E. Traub. On the equivalence of the traits assessed by multiple-choice and constructed-response tests. *Construction versus choice in cognitive measurement: Issues in constructed response, performance testing, and portfolio assessment*, pages 29–44, 1993.

[32] W. J. Van der Linden and R. Hambleton. Handbook of item response theory. *Taylor & Francis Group. Citado na pág*, 1(7):8, 1997.

[33] F. Wang, Q. Liu, E. Chen, Z. Huang, Y. Chen, Y. Yin, Z. Huang, and S. Wang. Neural cognitive diagnosis for intelligent education systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6153–6161, 2020.

[34] F. Wang, Q. Liu, E. Chen, Z. Huang, Y. Yin, S. Wang, and Y. Su. Neuralcd: A general framework for cognitive diagnosis. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[35] X. Wang, C. Huang, J. Cai, and L. Chen. Using knowledge concept aggregation towards accurate cognitive diagnosis. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2010–2019, 2021.

[36] Z. Wang, Y. Gu, A. Lan, and R. Baraniuk. Varfa: A variational factor analysis framework for efficient bayesian learning analytics. *arXiv preprint arXiv:2005.13107*, 2020.

[37] L. Yao and R. D. Schwarz. A multidimensional partial credit model with associated item and test statistics: An application to mixed-format tests. *Applied psychological measurement*, 30(6):469–492, 2006.

[38] B. Zylich and A. Lan. Linguistic skill modeling for second language acquisition. In *LAK21: 11th International Learning Analytics and Knowledge Conference*, pages 141–150, 2021.

Unfolding Learners’ Response to Different Versions of Automated Feedback in a MOOC for Programming – A Sequence Analysis Approach

Hagit Gabbay, Anat Cohen
Tel Aviv University
hagitgabbay@mail.tau.ac.il anatco@tauex.tau.ac.il

ABSTRACT

In MOOCs for programming, Automated Testing and Feedback (ATF) systems are frequently integrated, providing learners with immediate feedback on code assignments. The analysis of the large amounts of trace data collected by these systems may provide insights into learners' patterns of utilizing the automated feedback, which is crucial for the design of effective tools and maximizing their potential to promote learning. However, data-driven research on the impact of ATF on learning is scarce, especially in the context of MOOCs. In the current study, we combine a theoretical framework of feedback with educational data mining methods to investigate the effect of feedback characteristics on learning behavior in a MOOC. Sequence pattern analysis is implemented to explore and visualize the actions taken by learners in response to feedback which composed of cognitive, meta-cognitive, and motivational elements. We applied our research approach in an empirical design which consists of five cohorts (total over 2200 learners) utilizing different versions of ATF. The findings suggest that learners tend to adopt learning strategies in response to feedback and exhibit a preference for utilizing example solutions, while still coping with the challenge of solving the assignments independently. The impact of feedback function, content and structure is discussed in light of a detailed view of the differences as well as common trends in learning paths. Allowing for fine-grained insights, we found our research approach contributes to a more comprehensive understanding of the effect of automated feedback characteristics in MOOCs for programming.

Keywords

Automated feedback, MOOCs for programming, educational data mining, sequence pattern analysis

1. INTRODUCTION

Automated Testing and Feedback (ATF) systems, integrated in programming courses, provide learners with immediate feedback on code assignments and allows for unlimited resubmissions. Research suggests that incorporating an ATF system into a programming course is perceived by learners as beneficial for their learning and motivation [2, 16]. Yet, research on the system’s effect on overall course outcomes has yielded inconclusive results and studies identified the main impact of the system in task-level (i.e. throughout solving code assignments) [1, 5, 20]. This may be a result of the complex nature of the feedback's effect, which is multifaceted and contingent upon various factors, including the feedback design [32]. In Massive Open Online Courses (MOOCs) for programming, characterized by large numbers of learners and self-directed learning, there is potential for ATF systems to assist learners and compensate for the lack of available instructor support. To design the automated feedback in MOOCs to maximize its effectiveness, it is necessary to understand how learners utilize the feedback and how feedback features affect learning. However, there is a lack of empirical studies on the impact of automated feedback characteristics on learning [20, 50], particularly in the context of MOOCs.

The asynchronous and self-paced nature of MOOCs poses challenges for instructors seeking to monitor and evaluate learners' utilization of the ATF system. In particular, it is difficult to determine the effectiveness of the feedback elements. Analyzing the large amounts of trace data collected by ATF systems may provide insights into learners' patterns of utilizing the feedback. However, data-driven research on the impact of automated feedback features on learning in MOOCs is scarce.

Addressing these gaps, the aim of the current study is to explore the effects of automated feedback characteristics on learning behavior in a MOOC for programming. To do so, we compare the behavior patterns of learners utilizing different versions of an ATF system, composed of cognitive, meta-cognitive, and motivational elements, within a MOOC. A data-driven approach is employed, consists of sequence pattern mining and statistical analysis. Notable, the assessment of our research approach is another goal of this study.

Sequence pattern mining (SPM) is a prevalent method within the domain of educational data mining for uncovering patterns in the sequential interactions of learners with educational systems [3]. By identifying patterns in the order and timing of learners' actions, SPM can provide valuable insights into learning strategies and behavior, which can be used to adapt and improve educational environments [4]. Unlike other methods, such as process mining, sequence mining is particularly well-suited for high-resolution analysis of learning behavior "at the local level", such as solving assignments [8]. Previous studies applied SPM to analyze learners’ interaction with different course materials, examine learning behaviors during different periods of learning or identify different sequence patterns between predefined groups in different research conditions (e.g. [12, 39]). The process of solving a code assignment involves sequential actions taken by the learner over the time period allocated for the task. Characterized by order and timing, these sequences of actions reflect the pattern of interaction between the learner and the ATF system. Given our goal to compare behavior patterns in response to different versions of the ATF, the SPM method may be a suitable and applicable approach.

H. Gabbay and A. Cohen. Unfolding learners’ response to different versions of automated feedback in a mooc for programming – a sequence analysis approach. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 160–170, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115711

2. RELATED WORK

2.1 Research framework for feedback effectiveness

The framework proposed by Narciss suggests that feedback can be characterized by three key factors, namely its function, content and presentation, all of which impact its effectiveness [32]. The function of feedback corresponds to the facet(s) of competencies it seeks to enhance, and can be classified as cognitive, meta-cognitive and motivational [33]. Cognitive feedback is aimed at promoting high-quality learning outcomes and the acquisition of the knowledge and cognitive operations necessary for accomplishing learning tasks (e.g. [31]). Metacognitive feedback directs the student's awareness of and ability to choose appropriate learning strategies [32], while motivational feedback may encourage students in maintaining their effort and persistence [33].

Feedback content is the information provided to the learner, which addresses the selected function. The content varies in terms of level of detail, as classified by [32]: basic feedback which only provides knowledge of result (KR), and sometimes knowledge of the correct response (KCR), or elaborate feedback (EF) providing additional information. [21] specified subtypes of elaborated feedback to classify the content of feedback on programming assignments, which can be identified in ATF systems. Among these, the relevant types for the current study include knowledge about mistakes (KM), such as test-failure errors and compiler errors; knowledge about metacognition (KMC), which relates to metacognitive feedback; and knowledge about how to proceed (KH), such as hints or examples. In the current study, we refer to the included components (e.g. text, hints or examples) as feedback structure. The content and structure both convey feedback function(s).

The presentation of feedback pertains to the way in which the content is presented or communicated to the learner, e.g. the timing, number of attempts, adaptability, or modality [32]. In ATF systems the feedback is commonly immediate, while the allowed number of attempts and the level of adaptivity, as well as the visuality of the provided feedback, may vary according to the system's characteristics and pedagogical approach [21, 35].

2.2 The impact of Feedback features

In the study in hand the focus is on the impact of the function and content of feedback, with no consideration of the presentation factor. Therefore, we consider feedback’s features as the function, content, and structure.

In a comprehensive literature review, [21] revealed that most systems provide cognitive feedback, typically in the form of information about errors and, occasionally, guidance on how to progress. In studies comparing the effectiveness of different types of feedback, [19, 20] suggest that correct/incorrect feedback (KR) is relatively less effective compared to KCR and EF. Their findings revealed that students who were provided with higher levels of feedback (KCR, EF) outperformed those who received only KR feedback across three complex programming assignments. Furthermore, the provision of KR feedback caused learners to make more attempts per assignment.

The impact of providing hints and/or example solutions, in various forms, has been examined in several studies. [19, 50] have found that providing students with fixed content hints as a help option has no significant impact on problem-solving performance. On the other hand, [28, 29] suggested that adaptive next-step on-demand hints have a positive effect on students' performance and learning. Yet, providing adaptive hints could be either computationally expensive or require great attention from human instructors and it should be considered whether it is cost-effective [20]. Example solutions, provided to learners in the form of hints during various stages of the problem-solving process, were indicated as more effective than other forms of feedback by [1, 34, 50]. The question of how learners react when they are given a choice between several help options has not yet been sufficiently investigated [18].

Metacognitive feedback was found only in few ATF tools [21], with inconclusive effects. [46] have shown positive results in the effect of metacognitive feedback on the strategy of collaborative learning. In contrast, [13] investigated the effect of explainable feedback on changes in learning strategies but no significant effect was found. A study in different knowledge domain found that immediate metacognitive feedback can help students acquire better help-seeking skills within an intelligent tutoring system for geometry. Moreover, the improved help-seeking skills transferred to learning new domain-level content [41]. Despite the limited research on the effect of metacognitive feedback in ATF, developing learning strategies is crucial for MOOC learners [44] thus we find it worth exploring this approach further.

Motivational feedback is also less common and in certain systems it combined with other forms of feedback [7]. Studies have demonstrated an effective use of motivational feedback, provide students with immediate positive feedback on completed objectives [45] or supportive motivational messages triggered by log data analysis [37]. [31] suggest a positive impact of motivational automated feedback on student engagement and performance in another knowledge domain.

In many cases, the feedback provided by ATF systems is not unidimensional and consists of more than one function, as well as a combination of several types of information [22]. Therefore, to investigate the impact of specific features, a comparative study is necessary, in which multiple versions distinctly differentiated in their features, are implemented simultaneously under consistent conditions pertaining to both the learning environment and the learners. With this approach and based on the proposed framework, we examined the impact of automated feedback features on learners' behavior patterns throughout solving code assignments. Of particular interest were the effects of detailed knowledge about mistakes (KM), hints, and example solutions (KH), as well as the provision of metacognitive and motivational functions which were less investigated.

2.3 Sequential pattern analysis of learning behavior

A variety of studies have been conducted to analyze the educational behaviors of students using sequence mining methods [43]. The objectives and applications of these studies vary widely, as well as the analytic approaches. For example, [24] conducted an analysis of MOOC log data of learners who completed final assessments, with the aim of comparing the behavioral patterns of learners with varying levels of achievement. The study employed SPM to extract frequencies of predefined sequences, representing engagement and time management behaviors. Subsequently, statistical analysis was performed to uncover distinctions among the groups. [23] utilized sequence mining techniques to identify differentially frequent patterns between distinct groups of students, without predefining patterns of learning. By utilizing performance data, segments of productive and unproductive learning behaviors were identified and compared. [4] analyzed data of MOOC for learning programming principles, to investigate study patterns exhibited by learners during assessment periods and the evolution of these patterns over time. Sequence mining methods were applied in two approaches, with predefined patterns and in an unsupervised manner, to capture study patterns from learners' interaction sequences.

For more comprehensive examination, recent studies use sequence pattern analysis in conjunction with other EDM methods (e.g. process mining) to identify and investigate learning tactics and strategies [3, 11, 25, 40] or to construct prediction models for learners' behavior such as dropout or course completion [8]. In order to gain insights of the way learners utilize ATF system, [30] analyzed records of submitted code assignments and clustered programs with similar functionality. SPM was then applied to trace student progress throughout an exercise. However, the effect of feedback characteristics on learners' behavior was not explored.

3. RESEARCH QUESTIONS

The present study aims to address the research gaps concerning the effect of automated feedback features in the context of MOOC for programming. With a data-driven approach, we employ sequence pattern analysis to explore and compare learners' response to various forms of feedback, which differ in terms of feedback features: function, content and structure. In this regard, the following research questions were posited:

RQ1: How do feedback features affect learners' behavior patterns throughout solving code assignments in a MOOC for programming? In particular,

RQ2: What is the impact of feedback features on the usage of the various help forms?

RQ3: What are the characteristics of utilizing hints and example solutions, and to what extent do they contribute to advancing the correct solution?

In an implementation perspective, our research approach may provide instructors with fine-grained insights of the utilization of ATF tools in MOOCs for programming, in order to maximize their contribution to learning. Therefore, the assessment of this approach is another goal of this study.

4. METHODOLOGY

To explore the connections between feedback features and learners' response we integrated an ATF system into a MOOC for programming and developed five different versions of the feedback provided by the system.

4.1 The ATF system

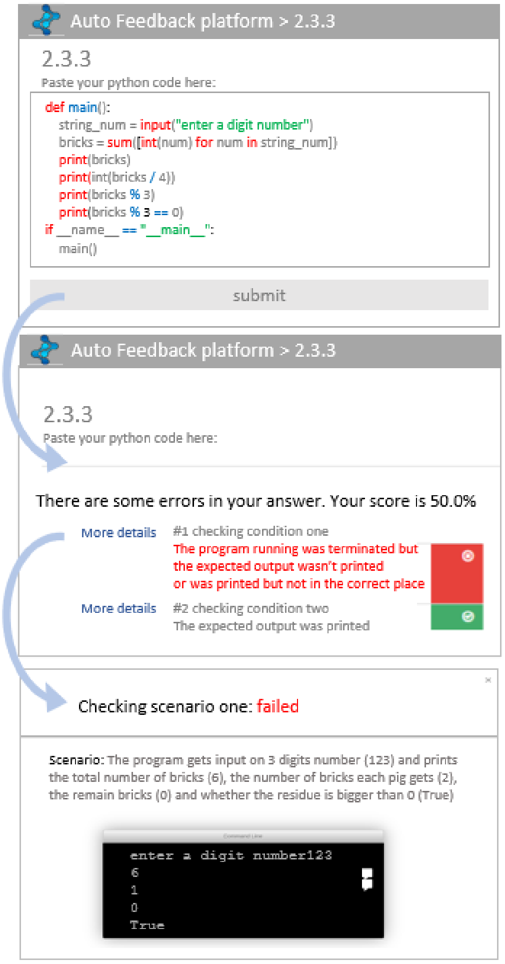
The integrated system is INGInious – an open-source software, supporting several programming languages and suitable for online courses (for more details see [10]). Applying static checks and running pre-defined tests for each assignment, the system provides immediate feedback, consists of a grade and a textual message. Error types detected are syntax errors, incorrect implementation of instructions, exception errors (prevent the code from being executed) and test-failed errors, where the results do not match the expected ones. For each type of error in each assignment, the feedback can be customized in advance to include an appropriate text and additional objects such as hints or example solution. In the current study, we used this configurability to explore learners' behavior in response to different feedback versions.

4.2 Feedback Versions

Based on the framework suggested by Narciss [32], five versions of the ATF system were designed, each with a different feedback function, content and structure. We tailored the feedback for the various error types to reflect the differences between the versions:

(1) **The Base version (V-Base)** provides the compiler message as-is for syntax errors. For exception and test-failed errors, the feedback includes a description of what should have been executed or output. An optional help form, which appears as "More details" (referred to as HMD from here on), is available. It consists of the exact breakdown of the actual vs. expected outcome (Figure 1). Although this version can be classified as cognitive feedback [32], we consider it "cognitive-light", compared to the other versions.

(2) **The Enriched Cognitive version (V-EC)** provides more elaborate text for each error type, offering cognitive knowledge. In addition to the HMD, a hint is also available upon request (i.e. clicking on a link). The hints are predefined and formulated based on common errors for each assignment (similar to [19]). They guide towards the correct solution but are not adaptive. In case of in case of multiple requests, the same text is presented for the same assignment each time.



Auto Feedback platform > 2.3.3

2.3.3

Paste your python code here:

```
def main():
    string_num = input("enter a digit number")
    bricks = sum([int(num) for num in string_num])
    print(bricks)
    print(int(bricks / 4))
    print(bricks % 3)
    print(bricks % 3 == 0)
if __name__ == "__main__":
    main()
```

Auto Feedback platform > 2.3.3

2.3.3

Paste your python code here:

There are some errors in your answer. Your score is 50.0%

More details #1 checking condition one
The program running was terminated but the expected output wasn't printed or was printed but not in the correct place

More details #2 checking condition two
The expected output was printed

Checking scenario one: failed

Scenario: The program gets input on 3 digits number (123) and prints the total number of bricks (6), the number of bricks each pig gets (2), the remain bricks (0) and whether the residue is bigger than 0 (True)

```
enter a digit number123
6
1
0
True
```

Figure 1. An example of the interface of the ATF system (V-Base version): the submitted code (top), the corresponding feedback message (middle), and the "More details" (HMD) option (bottom).

(3) **Meta-cognitive version (V-MC)** – knowledge of learning strategies, as defined by [41], is added to the text messages of V-EC. To enhance help-seeking strategies, learners are

encouraged to use the provided help forms and, in some cases, review the relevant content in the learning units. Additionally, after submitting a correct solution, an example solution is made available as a further learning strategy.

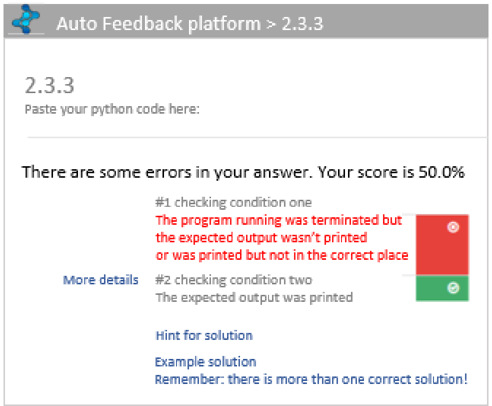
(4) **Motivational version (V-Motiv)** – feedback messages in V-EC are enhanced with positive and motivational language. Similar to [31], the text includes encouraging statements for partially correct solutions and overall motivating phrases.

(5) **Example solution version (V-ES)** - provides an option to view an example solution immediately following the initial attempt on an assignment, in addition to HMD and hints (Figure 2). Unlike other studies that recommend solutions for the next step or a similar assignment (e.g. [36, 47]), V-ES offers a complete solution for the current assignment.

Table 1 summarizes the functions and structures of the various versions and provides an example of the text presented.

Table 1. Function and structure of the five feedback versions

Version and function	Optional help forms	Examples of feedback text message (for test-failed errors) and hints
V-Base Cognitive "light"	HMD	*The program run was terminated but the expected output (…) wasn't printed or was printed but not in the correct place.*
V-EC Cognitive	HMD, hints	In addition to the text of V-Base: *The tested case is..* *The reason for the error may be.. or..* *Did you check…?* *Are the prints ordered correctly?*
		An example of a hint: *The input includes two types. Use if, else or elif to separate the input into two types (e.g. float/int or upper/lower case) and then run the appropriate conversion function.*
V-MC Cognitive + Meta-Cognitive	HMD, hints, example solution (Only after a correct solution)	In addition to the text of V-EC: *Use "More details" to check the received output.* Hints are the same as for V-EC.
V-Motiv Cognitive + Motivational	HMD, hints	In addition to the text of V-EC, if some case tests run correctly: *The program has shown to be successful in some test cases, indicating that you are headed in the right direction.* *Great job, you're making progress! Keep working on making the solution compatible with all input types and resubmit.* Hints are the same as for V-EC
V-ES Cognitive	HMD, hints, example solution	Text messages and hints are the same as for V-EC



2.3.3

Paste your python code here:

There are some errors in your answer. Your score is 50.0%

#1 checking condition one

The program running was terminated but the expected output wasn't printed or was printed but not in the correct place

More details

#2 checking condition two

The expected output was printed

Hint for solution

Example solution

Remember: there is more than one correct solution!

Figure 2. Enriched cognitive feedback with HMD, hint and example solution options, V-ES version.

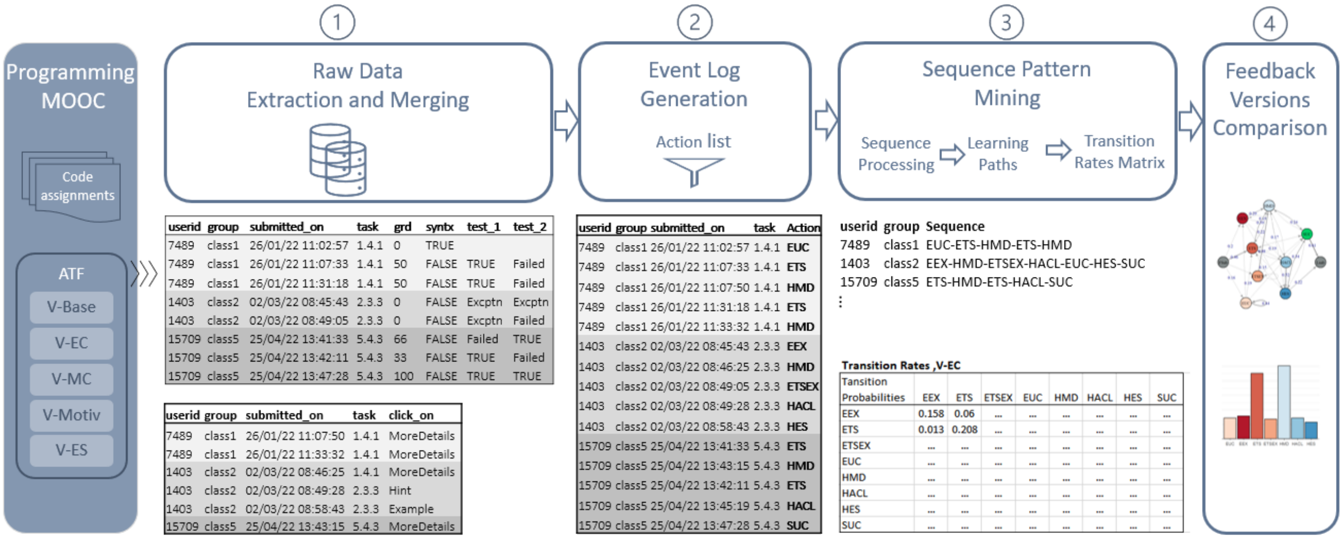
4.3 Research Field and data set

Our research field is a MOOC for learning the Python programming language, which is offered on an Edx-based platform. The course covers a range of topics organized into nine units, from the very basics of Python to the use of data structures, file manipulation, and functions. The course contains 29 video lectures, 39 exercises, and 53 code assignments, which are of varying levels of difficulty. It follows a self-paced learning mode, with all course materials available at once and no set deadlines. It is offered free of charge, although a certificate can be earned for a small fee. Learners interested must, in addition to paying the fee, complete 70% of the closed exercises and submit a concluding project.

The INGInious ATF system with the five feedback versions was incorporated into the course schedule between January to July 2022. It was incorporated as an external tool and configured to allow for unlimited submissions, aligning with MOOC learning concepts [35]. The cohort-mechanism embedded in the Edx platform was utilized to randomly assign learners enrolled in the course to one of five groups, each of which had access to a different feedback version. The allocation was carried out during registration and was fixed for the duration of the course, thus precluding any transfer between groups and ensuring that each learner was exposed exclusively to a single version of feedback.

The usage of the ATF system was voluntary. Of 16,602 enrollees, 2206 learners (13.3%) chose to register and use the system. Demographics provided by 75% of these learners, with 28% female, 71.5% male, and 0.5% "other". The age range varied from less than 11 years old to over 75 years old, with 11.17% under the age of 18, the majority (69.41%) between the ages of 18 to 34, and 19.42% above 34 years old. 58.7% had no prior knowledge in programming, 29.3% had knowledge in other programming languages, and 12% had prior knowledge in Python. A chi-squared test confirmed equal variance of gender, age and prior knowledge among the five experimental groups of the ATF users.

Data resources consist of ATF log files, including 165,282 submission records and 57,556 records of clicks on help forms offered in the various feedback versions ("help-clicks"). In this study, we analyzed a subset of the data, which only includes the actions of the learners when solving the four assignments in Unit 4 (11,519 submission and 8,769 help-clicks records). The research was conducted under the rules of ethics, while protecting privacy and maintaining the security of information, and in accordance with the approval of the university ethics committee.



userid	group	submitted_on	task	grd	syntx	test_1	test_2
7489	class1	26/01/22 11:02:57	1.4.1	0	TRUE		
7489	class1	26/01/22 11:07:33	1.4.1	50	FALSE	TRUE	Failed
7489	class1	26/01/22 11:31:18	1.4.1	50	FALSE	TRUE	Failed
1403	class2	02/03/22 08:45:43	2.3.3	0	FALSE	Excptn	Excptn
1403	class2	02/03/22 08:49:05	2.3.3	0	FALSE	Excptn	Failed
15709	class5	25/04/22 13:41:33	5.4.3	66	FALSE	Failed	TRUE
15709	class5	25/04/22 13:42:11	5.4.3	33	FALSE	TRUE	Failed
15709	class5	25/04/22 13:47:28	5.4.3	100	FALSE	TRUE	TRUE

userid	group	submitted_on	task	click_on
7489	class1	26/01/22 11:07:50	1.4.1	MoreDetails
7489	class1	26/01/22 11:33:32	1.4.1	MoreDetails
1403	class2	02/03/22 08:46:25	1.4.1	MoreDetails
1403	class2	02/03/22 08:49:28	2.3.3	Hint
1403	class2	02/03/22 08:58:43	2.3.3	Example
15709	class5	25/04/22 13:43:15	5.4.3	MoreDetails

userid	group	submitted_on	task	Action
7489	class1	26/01/22 11:02:57	1.4.1	EUC
7489	class1	26/01/22 11:07:33	1.4.1	ETS
7489	class1	26/01/22 11:07:50	1.4.1	HMD
7489	class1	26/01/22 11:31:18	1.4.1	ETS
7489	class1	26/01/22 11:33:32	1.4.1	HMD
1403	class2	02/03/22 08:45:43	2.3.3	EEX
1403	class2	02/03/22 08:46:25	2.3.3	HMD
1403	class2	02/03/22 08:49:05	2.3.3	ETSEX
1403	class2	02/03/22 08:49:28	2.3.3	HACL
1403	class2	02/03/22 08:58:43	2.3.3	HES
15709	class5	25/04/22 13:41:33	5.4.3	ETS
15709	class5	25/04/22 13:43:15	5.4.3	HMD
15709	class5	25/04/22 13:42:11	5.4.3	ETS
15709	class5	25/04/22 13:45:19	5.4.3	HACL
15709	class5	25/04/22 13:47:28	5.4.3	SUC

userid	group	Sequence
7489	class1	EUC-ETS-HMD-ETS-HMD
1403	class2	EEX-HMD-ETSEX-HACL-EUC-HES-SUC
15709	class5	ETS-HMD-ETS-HACL-SUC
⋮		

Transition Rates ,V-EC

Tansition Probabilities	EEX	ETS	ETSEX	EUC	HMD	HACL	HES	SUC
EEX	0.158	0.06	...	...	...	...	...	...
ETS	0.013	0.208	...	...	...	...	...	...
ETSEX	...	...	...	...	...	...	...	...
EUC	...	...	...	...	...	...	...	...
HMD	...	...	...	...	...	...	...	...
HACL	...	...	...	...	...	...	...	...
HES	...	...	...	...	...	...	...	...
SUC	...	...	...	...	...	...	...	...

Figure 3. Analytic approach to uncover learning patterns and compare the response to feedback among experimental groups.

4.4 Analytic approach to detect patterns of learning behavior

To explore learning behavior in fine-grained manner we utilized SPM, analyzing patterns of actions taken by learners throughout solving code assignments. Based on variables gathered from the sequence analysis, statistical tests were conducted to compare between the five experimental groups.

The method we applied to identify sequence patterns is similar to those described in previous studies [11, 43] (Figure 3). First, raw data of submission and help-click records for each learner were extracted and merged, based on assignment id and time stamps (1). Using a list of action codes, composed to represent learner actions (detailed below), we generated the action-event log (2). The exploratory sequence analysis implemented in the TraMineR package of R [15] was then utilized (3) to produce sequences of actions for each assignment and learner, as well as compute transition probability matrices and uncover characteristics of learning paths. Finally, we visualized and compared behavior patterns, using the five experimental groups as designated clusters (4).

4.4.1 Response Actions and definitions

Upon receiving feedback, a learner may resubmit a revised solution, use help, or waive the assignment without making any further submissions. To identify the differences between consecutive submissions, the "resubmitting" action is represented by several specific actions, based on error-types detected by the ATF system (section 4.1). The response-action list used for sequence analysis is described in Table 2.

4.4.2 Sequence processing

The action-event log, produced by converting the raw submission data, was transformed into sequences of actions for each assignment and each learner attempted. Sequences of length 1 (one submission to the assignment) were removed, having no evidence of feedback effect. The same applied to sequences including only repeated SUC actions.

Repeated identical actions of opening a hint or an example solution (HACL or HES) have been replaced with a single action, since the hints and example solution are fixed (for each assignment). To obtain a more representative dataset, very long sequences (higher than the 95th quantile) were removed. The response time for resubmission, defined as the period of time between consecutive submissions, was calculated for each pair of submissions. Response time longer than IDLE TIME (determined as 10 minutes) were replaced with IDLE TIME value.

Table 2. Response-action codes and definitions

Action type	Action code	Description
Submitting Actions	SUC	A correct solution.
	EUC	Code with syntax error.
	EIN	Code with instructions mismatch
	EEX	Code with exception for all test cases (can't be executed).
	ETS	Code that fails in some/all cases (the results differ from the expected ones)
	ETS-EX	Code with a mix of exceptions and test-failed errors
Help usage actions (help-clicks)	HMD	Clicking on "More details".
	HACL	Clicking on "hint"
	HES	Clicking to open the example solution.
"Fake" actions	Fsub, Lsub	First and last actions in sequence, added for visualization

4.4.3 Sequence analysis

A typical sequence consists of code-submission actions and help-usage actions (Figure 4). To explore the behavior patterns of learners, we analyzed the produced sequences from two perspectives: (1) The frequency of a specific action or a group of actions within the entire actions performed by a group of learners (e.g. the frequency of help-clicks in sequences 6-7 is 0.42) and (2) The percentage of learners who performed a specific action or subsequence of actions out of a group of learners (e.g. 29% of learners performed the EEX action (two out of seven)). This measure is the support of a pattern of actions.

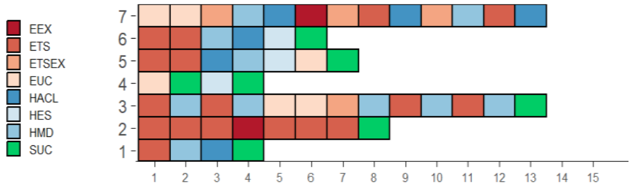


Figure 4. Example of sequences produced, representing the actions taken by learners when solving an assignment (each sequence represents the actions of a single learner).

5. RESULTS

In order to assess the effect of the feedback characteristics and evaluate the usability of our approach, we applied it to analyze the data of learners' interactions with the ATF system while completing four assignments in unit 4 of the course. These assignments are relatively homogeneous in terms of difficulty level and distribution of common error types, with a sufficient number of learners in each group who submitted more than once (a total of 567 learners and over 20,000 submission and help-click records). Data was analyzed for each assignment separately and for all four assignments together. We used non-parametric Kruskal-Wallis and Dunn tests to compare the relevant variables between groups, as the assumptions for one-way ANOVA were not met [6]. The reported p-values adjusted with the Bonferroni method.

5.1 RQ1: The Effect of Feedback Features on Learning Patterns

The analysis of learners' sequence of actions revealed significant differences in patterns between some of the experimental groups, as well as similarities among others. The average number of actions performed by learners (in all four assignments), represented by mean sequence length, was higher for groups V-MC and V-EC and lower for V-ES and V-BASE (Table 3).

Table 3. Sequence characteristics: length, time between actions and total duration (minutes). Higher values in red, lower values in blue.

Version (learners in group)	Sequence Length Mean (SD)	Time between actions, Mean (SD)	Time on Assign., Mean (SD)
V-Base (95)	10 (6.65) [1]	1.33 (0.91)	14.18 (13.5)
V-EC (112)	12.6 (7.92) [2]	1.40 (0.74) [3]	18.37 (14.74)
V-MC (116)	12.8 (8.45) [2]	1.31 (0.75)	17.45 (14.75) [4]
V-Motiv (121)	11.2 (8.81) [2]	1.44 (0.82) [3]	17.08 (14.84) [4]
V-ES (123)	9.51 (6.65) [1]	1.23 (0.79)	12.4 (11.30)
Kruskal-Wallis	$x^2(4) = 299.03$ $p<.001$	$x^2(4) = 141.34$ $p<.001$	$x^2(4) = 337.62$ $p<.001$

[1] no significant diff. between V-Base and V-ES
[2] no significant diff. between V-EC, V-MC and V-Motiv
[3] no significant diff. between V-EC and V-Motiv
[4] no significant diff. between V-EC and V-Motiv

The time intervals between actions, which may indicate the time spent processing feedback and revising the solution, were found to be longer, on average, for learners in V-EC and V-Motiv groups and shorter for learners in V-ES group. Additionally, the mean duration from first to last action in the sequence, representing time spent on the assignment, was highest among V-EC learners and lowest among V-ES learners. The distribution of the actions related to submitting assignments was consistent among all groups, as indicated by similar frequencies of EUC, EEX, ETS and ETS-EX detected in submissions. The percentage of learners who reached the correct solution, measured by the frequency of SUC action, was similar as well, and close to 100%. Thus, no difference was found regarding the score on assignments.

5.2 RQ2: The Impact of Feedback Features on the Usage of the Help Forms

Differences were observed between the groups in terms of help-usage actions. The variety of help forms offered to each group affected the overall use of help, as shown in Figure 5. With 0.34 of help-usage actions out of all actions, learners in the V-ES group used help more frequently compared to learners in the other groups ($x^2(4) = 24.78$, p<.001). V-Base showed the lowest usage of help, with 0.25 (only HMD in this case). Learners in V-EC and V-Motiv groups showed similar behavior in this manner, with 0.28 of help-use actions.

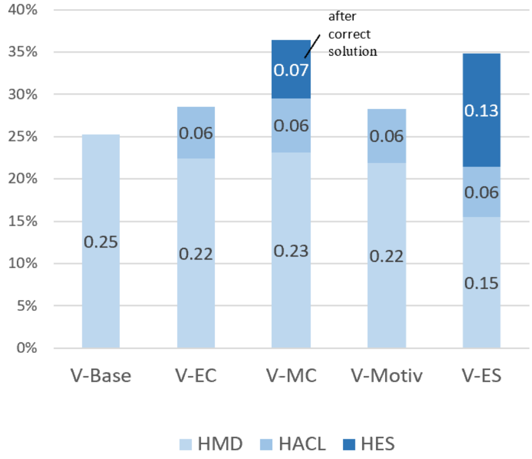


Figure 5. Frequency of help-usage actions performed by learners in each group

V-MC group utilized HES in 0.07 of actions but only after submitting a correct solution. Thus, their pattern of help-usage while attempting seems to be at medium level as well (0.29). Nevertheless, if we exclude these HES actions, the proportion of HMD actions during the solving process (before a correct solution is submitted) becomes 0.25, and the overall help usage of the V-MC group rises to 0.31. Although these adjusted values have not been found to be statistically significant, they suggest that learners in V-MC group tended to utilize the available help options (before submitting a correct solution) more frequently compared to V-EC and V-Motiv learners.

Upon further analysis, additional differences were found regarding the use of HMD, which is available in all versions in response to incorrect submissions. The likelihood of learners for this pattern of help seeking following EEX action reflects in the percentage of sequences containing the subsequence EEX-HMD at least once (Figure 6). The highest percent was found for V-MC group with 76% and the lowest for V-ES, with 43% (chi-squared independence, $x^2(4) = 14.664$, p<.01). The Pearson standardized residuals obtained (for using the HMD) are 2.39 for V-MC and -3.38 for V-ES. That is, learners provided with the metacognitive version responded to the explicit suggestion to use HMD after an exception error, as a learning strategy.

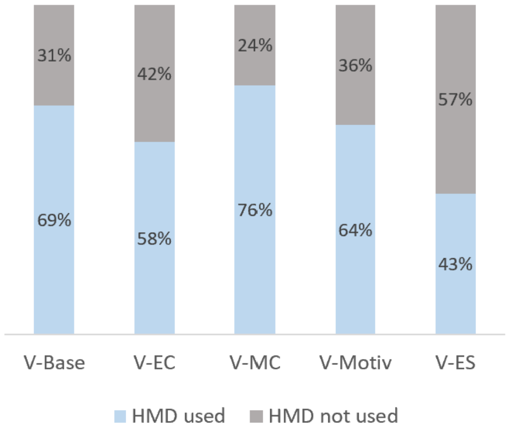


Figure 6. HMD usage after EEX (HMD-EEX proportional frequency)

Notably, the V-Base group showed close value of 69%, which is expected as HMD is the only help form provided in this version. A similar finding was obtained for the ETS-HMD sequence. The learners in the V-ES group were less likely to use HMD compared to the other groups while V-EC and V-Motiv groups exhibited comparable behavior.

5.3 RQ3: Patterns of Utilizing the Example Solutions and Hints

Calculating the probability matrix for transitions between states of each group allowed for detailed tracking of the use of help forms.

The example solution, as an aid to reach a correct solution, was offered only to learners in the V-ES group and was available right after the first submission attempt for an assignment. Approximately 75% of learners in this group utilized this form of help. Analysis of their action sequences revealed that the probability of transitioning **to** HES from one of the error states (EEX, ETS, ESTEX or EUC) was only about 0.34. This suggests that learners often first attempted to utilize other available forms of help, such as HMD or HACL, before turning to the example solution for assistance (Figure 7). **After** utilizing the example, the probability of resubmitting an incorrect code, signifying a transition from HES to one of the error states, was 0.31 (Figure 8). This outcome implies that in about one third of cases learners did not copy-paste the solution but tried to solve themselves. The likelihood of learners resubmitting a correct solution (i.e., transitioning from HES to SUC) was calculated to be 0.34, whereas the probability of seeking additional help (transitioning to HMD or HACL) was found to be 0.13. Only 0.22 of the cases resulted in learners choosing to waive and not resubmit. Taken together, these findings suggest that the availability of the example solution did not discourage learners in V-ES from trying to solve the assignment independently.

For V-EC learners, the example solution was offered as a mean of further learning, only becoming available after they submitted a correct solution. Therefore, it should not be considered as an aid to facilitate assignment completion. Approximately 58% of the learners in V-EC group chose to open the provided example.

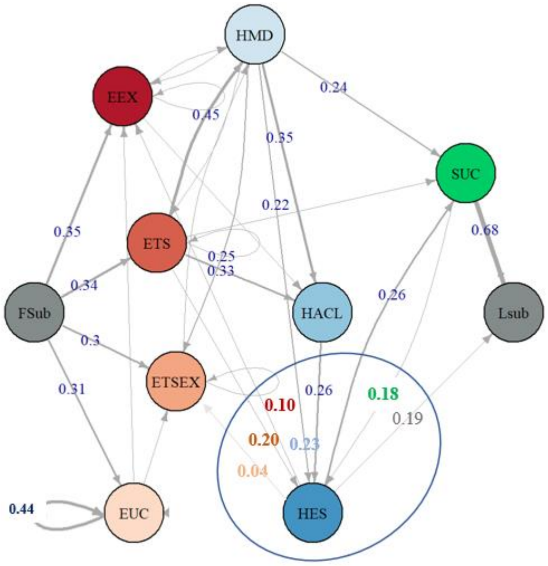


Figure 7. Transition probabilities for V-ES group. On edges: probability of transition to an action state (in particular, before opening HES). Probabilities less than 0.05 were omitted. Rates for HES are encircled and colored with the color of the "origin" state.

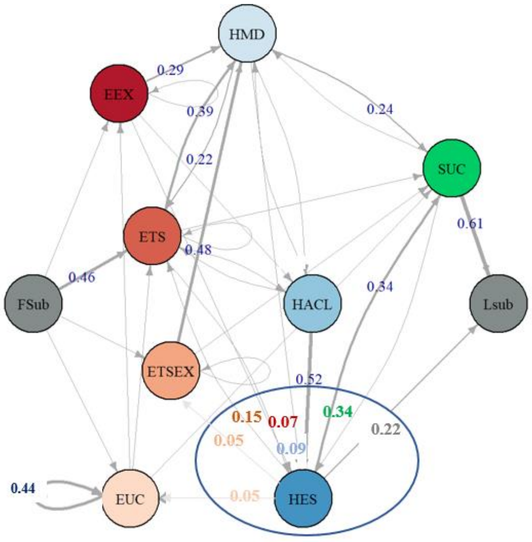


Figure 8. Transition probabilities for V-ES group. On edges: probability of transition from an action state (i.e. after opening HES). Probabilities less than 0.05 were omitted. Rates for HES are encircled and colored with the color of the "target" state.

The hint was provided to all groups of learners except V-Base. A quit low probability of 0.08 – 0.11 (with average of 0.09) was found for transitioning from EEX, ETS or ETSEX error-states to HACL, compared to 0.45-0.54 from any error-state to HMD (Figure 9). Furthermore, the probability the probability of transitioning to another form of help, either HMD or HES, after HACL, was found to be 0.39 on average. In contrast, the probability to submit a correct solution in the subsequent attempt following HACL was only 0.14, compared to 0.29 and 0.24 after HES and HMD, respectively. These findings indicate that the hint was less frequently requested and had a smaller impact on progress towards solving the assignments.

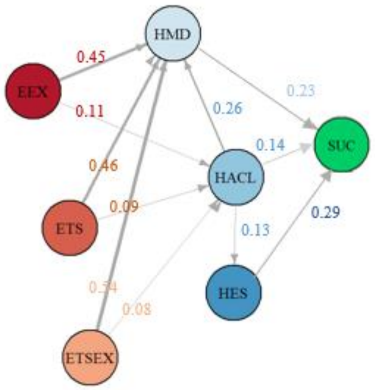


Figure 9. Transition probabilities to and from HACL, indicating low demand and low contribution.

6. DISCUSSION

In this empirical study, we implemented sequence pattern analysis to investigate the effects of automated feedback characteristics on the behavior patterns of learners, throughout solving code assignments within a MOOC for programming. For this purpose, five versions of ATF system were constructed, based on an initial version and distinctly differentiated by feedback function, message formulation, and help forms provided. Our analytic approach involved composing an "alphabet" of actions taken by learners, building learning paths and applying sequential pattern mining. Statistical tests were conducted to compare experimental groups.

Exploring learning paths revealed that integrating additional help resources to the feedback, in the form of hints and example solution, led to learners being more engaged in solving assignments by utilizing these helps. However, most learners in all the experimental groups persisted in submission attempts until arriving at the correct solution, leading to a "ceiling effect" on the achieved scores (which happens when a large percentage of observations score near the maximum grade on an assignment [28]). Here, the MOOC learning environment, characterized by unlimited attempts and the absence of knowledge evaluations, precludes an assessment of the impact of feedback structure on the level of knowledge acquisition.

The analysis did not reveal any significant impact of the motivational feedback, as the behavior of the learners in V-Motiv group was not distinct from that of learners provided with the enriched cognitive feedback (V-EC). As proposed by [27], motivational feedback may potentially have additional effects, such as on attitude and interest, but we did not collect data on these factors in the our study. Thus, among the findings of the current research, we would like to highlight learning behavior patterns exhibited by the group provided with metacognitive version (V-MC) and the group provided with the example solution version (V-ES), as the impact of these versions of feedback worth further investigation.

The metacognitive feedback is functioned to support learning through knowledge of learning strategies. In our design, the metacognitive version incorporates the strategies of help seeking and further learning through an example solution. Results suggest that to some extent, learners adopted both of these strategies. The encouragement to seek for more details about the error (HMD option) seems to affect V-MC learners towards utilizing it more often, compared to the other groups provided with the same help forms (i.e. V-EC and V-Motiv). It is worth noting that the instructions for using the ATF system, which were read by all learners *prior to starting*, include a general statement about the benefits of HMD. Thus, it is suggested the observed impact was made by the feedback, given during problem solving. Our findings support previous study, which identified a similar impact of metacognitive feedback on help-seeking behaviors, even with a long-term effect [41]. The importance of effective help-seeking in MOOCs and its association with better performance [26] highlight the potential positive contribution of metacognitive feedback to the learning process.

In addition, over half of the learners in the V-MC group utilized the example solution after submitting a correct code. This strategy has the potential to be an effective way of learning, as previously established by research [17]. [36] observed a similar engagement of students in an online course with solutions to code assignments they had already completed. Nevertheless, the motivation behind this behavior was based on prompts provided by the instructors. The ability to motivate learners in MOOCs to engage in "extra-learning" strategy through automated feedback, without requiring instructor involvement or summative assessment, is promising.

The learners in the group provided with example-solution version (V-ES) exhibited the most distinctive learning behavior. As expected, a majority of the learners in this group utilized the provided example solution, thereby reducing the time spent on assignments. Although they have made use of the other forms of help, it was to a lesser extent in comparison to the other groups. This finding is consistent with previous research, which has determined that a solution example is perceived by students as valuable, even more so than alternative forms of feedback [38].

However, the use of the provided example did not result in "help abuse" and did not discourage learners from attempting to independently tackle code writing. The learning path of most learners indicates efforts to solve the assignments (by utilizing other forms of help) **before** resorting to opening the solution. This behavior pattern is desirable, as research suggests that novice learners may benefit from actively engaging in solution attempts before they can make sense of given example [42]. Additionally, **after** utilizing the example, many learners did not demonstrate a pattern of copy-paste the solution, but continued to attempt the assignment, although fewer submissions were necessary to arrive at the correct solution. Nevertheless, providing the entire solution for a given assignment as a feedback form is an understudied area, in contrast to step-by-step examples (e.g., [49], [47]). Further research is necessary to examine its effectiveness and to better understand learners' perception of this type of feedback within the MOOC context.

The sequence analysis methods we applied facilitated a thorough examination of learners' utilization of the various help forms. The results indicate that the HMD was the predominant form of help, even when other forms of help such as hints or example solution were available. This finding suggests that KCR feedback, which includes only information about correctness and expected results, was not satisfying. Instead, learners sought for supplementary information. Previous studies support this finding by reporting of higher satisfaction exhibited by learners when provided with more detailed feedback for code errors [14]. On the other hand, [20] did not detect different attitudes of learners towards elaborated feedback, despite its demonstrated impact in improving performance. Additionally, unlike the study conducted by [48], our results do not provide any evidence of a connections between the use of elaborated feedback and increased engagement, as measured by the amount of time spent on completing assignments.

In line with the study of [20], hints were found as less prevalent form of help, as well as less effective, in comparison to HMD and example solution. One possible explanation is that the feedback consists of HMD is adaptive and tailored to the specific error detected, while hints in the implemented ATF are fixed and do not vary according to the submitted code. As a result, the feedback comprises HMD is geared towards rectifying identified errors, for the purpose of debugging, while the hints direct more to inquiring knowledge of concepts and found to be less useful by learners. Studies of learning environments with data-driven hints have showed different results, suggesting the use of hints shorten the time learners spent on task while achieving the same performance [38]. However, systems that include this type of support are more complex and creating the hints may be time-consuming for instructors [21]. Further study of the impact of hints' adaptivity on the extent of usage and effectiveness within MOOCs for programming is required to justify the investment of effort.

Our research approach allowed us to explore the learning behavior of students at a high resolution, identify patterns that were not observed with other tools, and compare between the experimental groups. However, we faced some challenges in utilizing this method. One of them was the significant computational time required to run multiple functions, such as searching for the most frequent sequences in each group, due to the relatively large number of learners and actions per sequence.

7. LIMITATIONS

Some limitations of this study need to be considered. First, data were collected in an authentic learning environment with low control on research setting and no indications of learners' behavior outside the course environment. In particular, other interpreters or automated feedback tools may be utilized to solve code assignments. The random assignment of learners to research groups may ensure equal tendency towards the use of such external tools, however, it has not been empirically validated.

Another limitation of this study is the narrow scope of the data analysis, which is restricted to four assignments that possess specific characteristics in terms of difficulty level and learning context. Feedback effects may vary with assignment features [32], therefore, generalizing the results of this study to diverse types of assignments should be approached with caution. Similarly, it would be beneficial to consider the features of the ATF system, such as the user interface and the manner in which various forms of help are presented (e.g. location on the screen or colors), as these factors may also have an impact on learners' interaction with feedback.

The method of SPM applied in this study had several shortcomings. Our approach involves predefining sequences of activities that represent behavioral patterns, and then analyzing their frequencies for each group. However, this method may not capture all significant differentiating patterns. An alternative approach, such as automatically capturing learning patterns from learners' interaction sequences, could potentially yield more informative findings that are relevant to the research questions. Additionally, the SPM method does not support the identification of start-to-end paths and thus the comparison between the experimental groups in terms of the entire process of solving the assignments was not allowed. A possible way to address the gap is to combine our method with process mining techniques, specifically Local Process Mining, suggested by [9], which may prove to be more compatible in the context of a single or group of assignments.

8. CONCLUSIONS AND FUTURE RESEARCH

The comparison of feedback versions in this empirical study adds to the research literature knowledge about the impact of different feedback characteristics, specifically in the context of MOOCs for programming. Significant results for our opinion, relevant to learning in MOOCs are (1) the possibility to influence learning strategies through targeted feedback function and (2) the indication for the deliberate use of example solutions by learners without negatively affecting their motivation to practice writing code themselves. These findings have implications for instructors in MOOCs, as they can use these insights to adjust the feedback provided in ATF systems to enhance support for MOOC learners. For example, to effectively encourage the use of additional help-seeking strategies such as consulting a discussion forum, or to provide additional examples of isomorphic assignments.

The data-driven approach can mitigate the gap of remote teaching and facilitate a process of on-going revising the automated feedback, by assessing the impact of changes and add-ons. Furthermore, instructors may detect problems within the assignments, by identifying, for example, patterns of repeated errors or high rates of waiving, and take steps to address these issues.

In conclusion, this study was focused on investigating the impact of feedback on learners' performance within the context of code assignments. A potential avenue for future research is to expand on this analysis and gain a more comprehensive understanding of the relationship between feedback characteristics and learning behavior. Such research may incorporate a combination of sequence and process mining methods to examine the entirety of the learners' engagement within the MOOC, including their interactions with course content (e.g., videos and comprehension exercises), through a compatible framework as previously proposed by [16].

With the increasing demand for programming skills in today's job market, MOOCs for programming have become an important tool for individuals looking to advance their careers or gain new ones, providing equal opportunities for programming education to a diverse audience. In addition, MOOCs are a valuable pedagogical supplement for instructors who seek to enhance their curriculum or provide supplementary resources for their students. We posit that this study, along with additional data-driven research in this domain, has the potential to foster the development of efficient ATF systems that promote learning in programming MOOCs, and consequently, enhance the success rate of a larger number of learners in these courses.

9. ACKNOWLEDGMENT

We express our gratitude to the Azrieli Foundation for awarding us a generous Azrieli Fellowship, which facilitated this research. Furthermore, we acknowledge our anonymous reviewers and the EDM program chairs for providing us with valuable feedback.

10. REFERENCES

[1] Ahmed, U.Z., Srivastava, N., Sindhgatta, R. and Karkare, A. 2020. Characterizing the pedagogical benefits of adaptive feedback for compilation errors by novice programmers. *Proceedings - International Conference on Software Engineering* (2020), 139–150.

[2] Benotti, L., Aloi, F., Bulgarelli, F. and Gomez, M.J. 2018. The effect of a web-based coding tool with automatic feedback on students' performance and perceptions. *SIGCSE 2018 - Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (2018), 2–7.

[3] Bogarín, A., Cerezo, R. and Romero, C. 2018. A survey on educational process mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. 8, 1 (Jan. 2018),

e1230. DOI= https://doi.org/10.1002/WIDM.1230.

[4] Boroujeni, M.S. and Dillenbourg, P. 2018. Discovery and temporal analysis of latent study patterns in MOOC interaction sequences. *ACM International Conference Proceeding Series* (Mar. 2018), 206–215.

[5] Cavalcanti, A.P., Barbosa, A., Carvalho, R., Freitas, F., Tsai, Y.-S., Gašević, D. and Mello, R.F. 2021. Automatic feedback in online learning environments: A systematic literature review. *Computers and Education: Artificial Intelligence*. 2, (Jan. 2021), 100027. DOI= https://doi.org/10.1016/J.CAEAI.2021.100027.

[6] Chan, Y. and Walmsley, R.P. 1997. Learning and understanding the Kruskal-Wallis one-way analysis-of-variance-by-ranks test for differences among three or more independent groups. *Physical Therapy*. 77, 12 (1997), 1755–1762. DOI= https://doi.org/10.1093/ptj/77.12.1755.

[7] Deeva, G., Bogdanova, D., Serral, E., Snoeck, M. and De Weerdt, J. 2021. A review of automated feedback systems for learners: Classification framework, challenges and opportunities. *Computers & Education*. 162, (Mar. 2021), 104094. DOI= https://doi.org/10.1016/J.COMPEDU.2020.104094.

[8] Deeva, G., De Smedt, J., De Koninck, P. and De Weerdt, J. 2018. Dropout prediction in MOOCs: A comparison between process and sequence mining. *Lecture Notes in Business Information Processing*. 308, January (2018), 243–255. DOI= https://doi.org/10.1007/978-3-319-74030-0_18.

[9] Deeva, G. and Weerdt, J. De 2018. Understanding Automated Feedback in Learning Processes by Mining Local Patterns. *Lecture Notes in Business Information Processing*. 342, (Sep. 2018), 56–68. DOI= https://doi.org/10.1007/978-3-030-11641-5_5.

[10] Derval, G., Gego, A., Reinbold, P., Benjamin, F. and Van Roy, P. 2015. Automatic grading of programming exercises in a MOOC using the INGInious platform.. *Proceedings of the European Stakeholder Summit on experiences and best practices in and around MOOCs (EMOOCS'15)* (Mons, Belgium, 2015), 91–86.

[11] Fan, Y., Tan, Y., Rakovi, M., Wang, Y., Cai, Z., Williamson Shaffer, D., Gaševi, D. and Yizhou Fan, C. 2022. Dissecting learning tactics in MOOC using ordered network analysis. *Journal of Computer Assisted Learning*. 39, 1 (Aug. 2022), 154–166. DOI= https://doi.org/10.1111/JCAL.12735.

[12] Faucon, L., Kidzí, Ł. and Dillenbourg, P. 2016. Semi-Markov model for simulating MOOC students. *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016* (2016), 358–363.

[13] Félix, E., Amadieu, F., Venant, R. and Broisin, J. 2022. Process and Self-regulation Explainable Feedback for Novice Programmers Appears Ineffectual. *Proceedings of the European Conference on Technology Enhanced Learning* (2022), 514–520.

[14] Finn, B., Thomas, R. and Rawson, K.A. 2018. Learning more from feedback: Elaborating feedback with examples enhances concept learning. *Learning and Instruction*. 54, (Apr. 2018), 104–113. DOI= https://doi.org/10.1016/j.learninstruc.2017.08.007.

[15] Gabadinho, A., Ritschard, G., Müller, N.S. and Studer, M. 2011. Analyzing and Visualizing State Se-quences in R with TraMineR. *Journal of Statistical Software*. 40, 4 (2011), 1–37. DOI= https://doi.org/10.18637/jss.v040.i04.

[16] Gabbay, H. and Cohen, A. 2022. Investigating the effect of automated feedback on learning behavior in MOOCs for programming. *EDM 2022 - Proceedings of the 15th International Conference on Educational Data Mining* (2022), 376–383.

[17] Garcia, R., Falkner, K. and Vivian, R. 2018. Systematic literature review: Self-Regulated Learning strategies using e-learning tools for Computer Science. *Computers and Education*. 123, (Aug. 2018), 150–163. DOI= https://doi.org/10.1016/j.compedu.2018.05.006.

[18] Gross, S. and Pinkwart, N. 2015. How do learners behave in help-seeking when given a choice? *Artificial Intelligence in Education: 17th International Conference, AIED 2015* (Madrid, Spain, 2015), 600–603.

[19] Hao, Q., Smith IV, D.H., Ding, L., Ko, A., Ottaway, C., Wilson, J., Arakawa, K.H., Turcan, A., Poehlman, T. and Greer, T. 2022. Towards understanding the effective design of automated formative feedback for programming assignments. *Computer Science Education*. 32, 1 (Jan. 2022), 105–127. DOI= https://doi.org/10.1080/08993408.2020.1860408.

[20] Hao, Q., Wilson, J.P., Ottaway, C., Iriumi, N., Arakawa, K. and Smith, D.H. 2019. Investigating the essential of meaningful automated formative feedback for programming assignments. *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC* (Oct. 2019), 151–155.

[21] Keuning, H., Jeuring, J. and Heeren, B. 2018. A systematic literature review of automated feedback generation for programming exercises. *ACM Transactions on Computing Education*. 19, 1 (2018), 1–43. DOI= https://doi.org/10.1145/3231711.

[22] Kiesler, N. 2022. An Exploratory Analysis of Feedback Types Used in Online Coding Exercises. *arXiv preprint* . 2022)).

[23] Kinnebrew, J.S., Loretz, K.M. and Biswas, G. 2013. A Contextualized, Differential Sequence Mining Method to Derive Students' Learning Behavior Patterns. *Journal of Educational Data Mining*. 5, 1 (2013), 190–219.

[24] Li, S., Du, J. and Sun, J. 2022. Unfolding the learning behaviour patterns of MOOC learners with different levels of achievement. *International Journal of Educational Technology in Higher Education*. 19, 1 (2022), 1–20. DOI= https://doi.org/10.1186/s41239-022-00328-8.

[25] Maldonado-Mahauad, J., Pérez-Sanagustín, M., Kizilcec, R.F., Morales, N. and Munoz-Gama, J. 2018. Mining theory-based patterns from Big data: Identifying self-regulated learning strategies in Massive Open Online Courses. *Computers in Human Behavior*. 80, (Mar. 2018), 179–196. DOI= https://doi.org/10.1016/j.chb.2017.11.011.

[26] Maldonado-Mahauad, J., Pérez-Sanagustín, M., Moreno-Marcos, P.M., Alario-Hoyos, C., Muñoz-Merino, P.J. and Delgado-Kloos, C. 2018. Predicting Learners' Success in a Self-paced MOOC Through Sequence Patterns of Self-regulated Learning. *Lifelong Technology-Enhanced Learning: 13th European Conference on Technology Enhanced Learning, EC-TEL 2018* (Leeds, UK, 2018), 355–369.

[27] Marwan, S., Fisk, S., Price, T.W., Barnes, T. and Gao, G. 2020. Adaptive Immediate Feedback Can Improve Novice Programming Engagement and Intention to Persist in Computer Science. *Proceedings of the 2020 ACM Conference on International Computing Education Research* (2020), 194–203.

[28] Marwan, S. and Price, T.W. 2022. iSnap : Evolution and Evaluation of a Data-Driven Hint System for Block-based Programming. XX, X (2022), 1–15. DOI= https://doi.org/10.1109/TLT.2022.3223577.

[29] Marwan, S., Williams, J.J. and Price, T. 2019. An evaluation of the impact of automated programming hints on performance and learning. *ICER 2019 - Proceedings of the 2019 ACM Conference on International Computing Education Research*. (Jul. 2019), 61–70. DOI= https://doi.org/10.1145/3291279.3339420.

[30] McBroom, J., Yacef, K., Koprinska, I. and Curran, J.R. 2018. A data-driven method for helping teachers improve feedback in computer programming automated tutors. *Artificial Intelligence in Education: 19th International Conference, AIED 2018, Proceedings, Part I* (London, UK, 2018), 324–337.

[31] Mitrovic, A., Ohlsson, S. and Barrow, D.K. 2013. The effect of positive feedback in a constraint-based intelligent tutoring system. *Computers and Education*. 60, 1 (Jan. 2013), 264–272. DOI= https://doi.org/10.1016/j.compedu.2012.07.002.

[32] Narciss, S. 2013. Designing and evaluating tutoring feedback strategies for digital learning environments on the basis of the interactive tutoring feedback model. *Digital Education Review*. 23, 1 (2013), 7–26. DOI= https://doi.org/10.1344/der.2013.23.7-26.

[33] Narciss, S. 2008. Feedback strategies for interactive learning tasks. *Handbook of research on educational communications and technology*. J.M. Spector, M.D. Merrill, J. Van Merriënboer, and M.P. Driscoll, eds. Taylor and Francis. 125–143.

[34] Perikos, I., Grivokostopoulou, F. and Hatzilygeroudis, I. 2017. Assistance and Feedback Mechanism in an Intelligent Tutoring System for Teaching Conversion of Natural Language into Logic. *International Journal of Artificial Intelligence in Education 2017 27:3*. 27, 3 (Feb. 2017), 475–514. DOI= https://doi.org/10.1007/S40593-017-0139-Y.

[35] Pieterse, V. 2013. Automated Assessment of Programming Assignments. *3rd Computer Science Education Research Conference on Computer Science Education Research*. 3, April (2013), 45–56.

[36] Price, T.W., Williams, J.J., Solyst, J. and Marwan, S. 2020. Engaging Students with Instructor Solutions in Online Programming Homework. *Conference on Human Factors in Computing Systems - Proceedings* (Apr. 2020), 1–7.

[37] Rajendran, R., Iyer, S. and Murthy, S. 2019. Personalized Affective Feedback to Address Students' Frustration in ITS. *IEEE Transactions on Learning Technologies*. 12, 1 (Jan. 2019), 87–97. DOI= https://doi.org/10.1109/TLT.2018.2807447.

[38] Rivers, K. 2018. Automated data-driven hint generation for learning programming. *Dissertation Abstracts International: Section B: The Sciences and Engineering*. 79, 4-B(E) (2018).

[39] Rizvi, S., Rienties, B., Rogaten, J. and Kizilcec, R.F. 2020. Investigating variation in learning processes in a FutureLearn MOOC. *Journal of Computing in Higher Education*. 32, (2020), 162–181. DOI= https://doi.org/10.1007/s12528-019-09231-0.

[40] Rohani, N., Gal, K., Gallagher, M. and Manataki, A. 2023. Discovering Students ' Learning Strategies in a Visual Programming MOOC through Process Mining Techniques. *Process Mining Workshops: ICPM 2022 International Workshops, Revised Selected Papers* (Bozen-Bolzano, Italy, 2023), 539–551.

[41] Roll, I., Aleven, V., McLaren, B.M. and Koedinger, K.R. 2011. Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learning and Instruction*. 21, 2 (Apr. 2011), 267–280. DOI= https://doi.org/10.1016/j.learninstruc.2010.07.004.

[42] Roll, I., Baker, R.S.J. d., Aleven, V. and Koedinger, K.R. 2014. On the Benefits of Seeking (and Avoiding) Help in Online Problem-Solving Environments. *Journal of the Learning Sciences*. 23, 4 (Oct. 2014), 537–560. DOI= https://doi.org/10.1080/10508406.2014.883977.

[43] Saint, J., Gaševic, D., Matcha, W., Uzir, N.A.A. and Pardo, A. 2020. Combining analytic methods to unlock sequential and temporal patterns of self-regulated learning. *ACM International Conference Proceeding Series*. (Mar. 2020), 402–411. DOI= https://doi.org/10.1145/3375462.3375487.

[44] Serth, S., Teusner, R. and Meinel, C. 2021. Impact of Contextual Tips for Auto-Gradable Programming Exercises in MOOCs. *Proceedings of the Eighth ACM Conference on Learning @ Scale* (New York, NY, USA, 2021), 307–310.

[45] Shabrina, P., Marwan, S., Chi, M., Price, T.W. and Barnes, T. 2020. The Impact of Data-driven Positive Programming Feedback: When it Helps, What Happens when it Goes Wrong, and How Students Respond. *Educational Data Mining in Computer Science Education Workshop@ EDM 2020* (2020).

[46] Vizcaíno, A. 2005. A Simulated Student Can Improve Collaborative Learning. *International Journal of Artificial Intelligence in Education*. 15, (2005), 3–40.

[47] Wang, W., Rao, Y., Zhi, R., Marwan, S., Gao, G. and Price, T.W. 2020. Step Tutor: Supporting Students through Step-by-Step Example-Based Feedback. *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*. (2020), 391–397. DOI= https://doi.org/10.1145/3341525.3387411.

[48] Wang, Z., Gong, S.Y., Xu, S. and Hu, X.E. 2019. Elaborated feedback and learning: Examining cognitive and motivational influences. *Computers and Education*. 136, (Jul. 2019), 130–140. DOI= https://doi.org/10.1016/j.compedu.2019.04.003.

[49] Zhi, R., Marwan, S., Dong, Y., Lytle, N., Price, T.W. and Barnes, T. 2019. Toward data-driven example feedback for novice programming. *EDM 2019 - Proceedings of the 12th International Conference on Educational Data Mining* (2019), 218–227.

[50] Zhou, Y., Andres-Bray, J.M., Hutt, S., Ostrow, K. and Baker, R.S. 2021. A Comparison of Hints vs. Scaffolding in a MOOC with Adult Learners. *Artificial Intelligence in Education: 22nd International Conference, AIED 2021, Proceedings, Part II* (Utrecht, The Netherlands, Jun. 2021), 427–432.

Can the Paths of Successful Students Help Other Students With Their Course Enrollments?

Kerstin Wagner
BHT* Germany
kerstin.wagner@bht-berlin.de

Agathe Merceron
BHT* Germany
merceron@bht-berlin.de

Petra Sauer
BHT* Germany
sauer@bht-berlin.de

Niels Pinkwart
DFKI† Germany
niels.pinkwart@dfki.de

ABSTRACT

In this paper, we present an extended evaluation of a course recommender system designed to support students who struggle in the first semesters of their studies and are at risk of dropping out. The system, which was developed in earlier work using a student-centered design and which is based on the explainable k-nearest neighbor algorithm, recommends a set of courses that have been passed by the majority of the student's nearest neighbors who have completed their studies. The present evaluation is based on the data of students from three different study programs. One result is that the recommendations do lower the dropout risk. We also discovered that while the recommended courses differed from those taken by students who dropped out, they matched quite well with courses taken by students who completed the degree program. Although the course recommender system targets primarily students at risk, students doing well could use it. Furthermore, we found that the number of recommended courses for struggling students is less than the number of courses they actually enrolled in. This suggests that the recommendations given indicate a different and hopefully feasible path through the study program for students at risk of dropping out.

Keywords

Course recommender system, nearest neighbors, explainability, user-centered design, dropout prediction

1. INTRODUCTION

In the last decades, universities worldwide have changed a lot. They offer a wider range of degree programs and courses and welcome more students from diverse cultural backgrounds. Further, teaching and learning at school differs from teaching and learning at university. Some students cope well and keep the same academic performance level at university as at school. Others struggle, perform worse, and might become at risk of dropping out.

The preliminary exploration of our data has shown, that most of the students drop out during the first three semesters of their studies. Therefore, the course recommendations proposed in this work focus on supporting struggling students after their 1st and 2nd semesters. The final goal in developing such a system is to integrate it into novel facilities that universities may set up to support their diverse students better.

At the beginning of each semester in Germany, students must decide which courses to enroll in. When entering university directly after high school for their 1st semester, most of them decide to enroll in exactly the courses planned in the study handbook. The decision becomes more difficult when they fail courses in their 1st semester and should choose the courses to enroll in their 2nd semester: should they repeat right away the courses they failed? Which courses planned in the 2nd semester in the study handbook should they take? Should they reduce the number of courses they enroll in to have a better chance of passing them all? Should they take more courses to compensate for the courses they failed? The study handbook does not help answer these questions.

Previous research has shown that most students rely on friends and acquaintances as one source of information when deciding which courses to enroll in [19]. Further, students wish to have explanations if courses are recommended to them. The present recommender system supports students in choosing which courses to take before the semester begins and is based on the explainable algorithm k-nearest neighbors (KNN). It recommends to students the set of courses that the majority of their nearest neighbors, who successfully graduated, have passed.

Nearest neighbors are students who, at the same stage in their studies, have failed or passed almost the same courses with the same or very similar grades. The system does not recommend top n courses as other systems do, e.g. [10, 12, 14, 15]. Rather, it recommends an optimal set of courses, and we assume that a student should be able to pass all the courses of that set. Because the recommendations are driven by past records of students who graduated, we also pose the hypothesis that students following the recommendations

[*] Berliner Hochschule für Technik

[†] Deutsches Forschungszentrum für Künstliche Intelligenz

K. Wagner, A. Merceron, P. Sauer, and N. Pinkwart. Can the paths of successful students help other students with their course enrollments? In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 171–182, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115719

should have a lower risk of dropping out. Using historical data, we evaluated the recommendations given after the 1st and 2nd semester. Although the recommendations are designed to support struggling students, every student should have access to them. The recommendations should show a different, more academically successful way of studying for struggling students and therefore differ from the courses that they pass or enroll in.

More precisely, this paper addresses the following research questions:

1. Do the recommendations lower the risk of dropping out?
2. How large is the intersection between the set of courses recommended and the set of courses a student has passed?
3. a) How many courses are recommended and b) does this number differ from the number of courses passed and enrolled in by students?

This work builds on our previous work [20] by using a larger dataset with three different study programs instead of one to answer research questions 1 and 2, and by adding research question 3 to further investigate the provided recommendations. For all three questions, it is relevant whether there is a difference between students with difficulties and students with good performance as well as between study programs and semesters.

The paper is organized as follows. The next section describes related works. In the third section, we present our data, and in the fourth section our methodology. The results and their discussion are presented in section 5. In section 6, we describe a preliminary evaluation with students. The last section concludes the paper and presents future works. To make this paper self-contained, sections 3 and 4 repeat some descriptions and explanations already presented [20].

2. RELATED WORK

Dropout Prediction. Since our work aims to support students at risk of dropping out, it is necessary for us to be able to assess students' risk. Researchers have used various data sources, representations, and algorithms to address the task of predicting dropout. Academic performance data quite often form the basis; adding demographic data does not inherently lead to better results [2] but has been done for example in [1, 2, 9]. The data can be used as is as features or aggregated into new features. In terms of the algorithms used for dropout prediction, they range from simple, interpretable models such as decision trees, logistic regression, and KNN [1, 2, 9, 21] to black-box approaches like AdaBoost, random forests, and neural networks [1, 2, 11] — there is no algorithm that performs best in all contexts. Because the current study examines the impact of course recommendations on predicted risk, we only use courses and their grades as features when performing dropout prediction in section 4.2.

Course Recommendations. Various approaches to course recommendation have been explored in recent years. Urdaneta-Ponte et al. provided an overview of 98 studies published between 2015 and 2020 and related to recommender systems in education [18]. They answered the questions, among others, about what items were recommended and for whom the recommendations were intended. Course recommendations were found to be the second most common research focus, with 33 studies after learning resources, and 25 of these papers targeted students. Ma et al. first conducted a survey to identify the factors that influence course choice [10]. Based on this, they developed a hybrid recommender system that integrates the aspects of interest, grades, and time into the recommendations. The approach was evaluated with a dataset containing the results of 2,366 students from 5 years and from 12 departments. They obtained the best results in terms of recall when all aspects are included but with different weights. Morsy and Karypis analyzed their approaches to recommend courses in terms of their impact on students' grades [12]. Based on a dataset that includes 23 majors with at least 500 graduated students from 16 years, they aim to improve grades in the following semester without recommending easy courses only. Elbadrawy and Karypis investigated how various student and course groupings affect grade prediction and course recommendation [6]. The objective was to make the most accurate projections possible. Around 60,000 students and 565 majors were included in the dataset. The list of courses from which recommendations were derived was pre-filtered by major and student level. This limitation is comparable to our scenario, in which students choose courses depending on their study program. None of these works has the aim of supporting struggling students when enrolling in courses.

Our contribution. The idea of building a recommender system to support struggling students in their course enrollment, based on the paths of fellow students with the potential of providing explanations came out of the insights gained from a semi-structured group conversation with 25 students [19]. We propose a novel, thorough approach to evaluate such a recommender system that includes the following characteristics:

- Studies have shown that course recommendations can have an impact on students' performance. However, students at-risk were not in focus. We employ a two-step dropout risk prediction to determine whether the recommendations reduce dropout risk.
- We recommend a set of courses, not top n courses; therefore we evaluate not only that the passed courses contain the recommended courses — similar to other evaluations [6, 10, 12] — but also that the recommended courses contain the courses students have passed using F_1 score.
- We evaluate whether the number of courses is adequate.

3. DATA

Data from three six-semester bachelor programs at a medium-sized German university were used to develop and evaluate the course recommender system: Architecture (AR), Computer Science and Media (CM), and Print and Media Technology (PT). These three programs differ not only in their topic but also in the number of students enrolled. The initial dataset included 3,475 students who began their studies between the winter semester of 2012 and the summer semester of 2019. We only used data about academic performance: students' course results from the first three semesters accounted for 45,959 records of information about course enrollments and exam results over the mentioned period. The

Table 1: Number of students by program P (AR, CM, PT), train and test data set (Type), and student status (D, G). The proportion of dropouts in the test dataset is used as risk indicator (Risk).

P	Type	D	G	Sum	Risk
AR	**Train**	91	371	462	0.197
	Test	43	73	116	0.371
CM	**Train**	154	267	421	0.366
	Test	67	39	106	0.632
PT	**Train**	37	171	208	0.178
	Test	21	32	53	0.396
AR + CM + PT		413	953	1,366	0.302

Table 2: Academic performance overview by program and semester (PS), and student status (D, G): mean number of courses enrolled in (MeanE), mean number of courses passed (MeanP), difference (Diff) between MeanE and MeanP, and mean grade (MeanGr).

	MeanE		MeanP		Diff		MeanGr	
PS	**D**	**G**	**D**	**G**	**D**	**G**	**D**	**G**
AR1	4.9	5.0	3.2	4.7	1.7	0.3	2.8	2.1
AR2	5.5	5.8	3.0	5.1	2.5	0.8	3.0	2.3
AR3	5.1	5.9	1.9	5.4	3.2	0.5	3.2	2.2
CM1	4.9	5.1	2.9	4.8	2.0	0.3	3.0	2.1
CM2	5.2	5.8	2.1	5.0	3.1	0.8	3.0	2.3
CM3	4.7	5.8	1.3	5.0	3.5	0.9	3.2	2.1
PT1	5.8	6.0	4.3	5.8	1.5	0.3	2.5	2.0
PT2	5.7	5.5	2.5	4.9	3.2	0.6	2.9	1.9
PT3	6.1	6.4	2.3	5.5	3.8	0.9	3.1	2.0

grading scale is [1.0, 1.3, 1.7, 2.0, 2.3, 2.7, 3.0, 3.3, 3.7, 4.0, 5.0], with 1.0 being the best, 4.0 being the worst (just passed), and 5.0 means fail. Students may enroll in courses without taking the exam. In this case, they do not receive a grade, but the enrollment is recorded. To graduate, students must pass all mandatory courses as well as a program-specific number of elective courses. The study handbook includes a suggested course schedule for the six semesters, which students may or may not follow. At any time in their studies, students are allowed to choose courses from all offered courses.

Outliers. We removed three types of students: A) outliers in terms of the number of passed courses based on the interquartile range. Indeed, students can receive credit for courses completed in previous study programs; in our data, these credits are not distinguishable from credits earned by enrolling in and passing a course but they may result in a large number of courses passed, far more than anticipated in the study handbook. We remove these outliers because they might impact negatively dropout prediction [13]. B) students who were still studying at the time of data collection since they can not be used to predict the risk of dropping out. C) students without at least one record (passed, failed, or enrolled but have not taken the exam) in each of the first three semesters.

Datasets. The final dataset included 1,366 students who either graduated (”graduates”, status G) or dropped out (”dropouts”, status D). For the programs AR and CM, we had similarly sized data sets with 578 and 527 students, but only 261 students for the PT program because it has fewer students, see programs AR, CM, and PT, rows train and test column Sum in Table 1. For dropout risk prediction, described later in section 4.2, the data sets were sorted by the start of the study and split into 80% training data, row train in Table 1, and 20% test data, row test in Table 1, so that prediction evaluation was done based on students who started their studies last. We call the proportion of dropouts in each data set ”dropout risk”, see column Risk in Table 1. For example, the dropout risk of the train set of the program Architecture AR is 0.20= 91/462. Table 2 provides an overview of the number of courses students enroll and pass on average, the difference between the number of courses enrolled and passed, and the average grade based on courses passed and failed, by program, semester, and student status. For example, students in program AR who dropped out in the first semester enrolled in 4.9 courses but passed 3.2 courses on average, and got an average grade of 2.8, whereas students who graduated enrolled in 5.0 courses and passed 4.7 courses on average, and got an average grade of 2.1. One notices that students with status D pass fewer courses per semester and receive lower grades.

Missing values. For the algorithms used for the recommendations and dropout predictions, we had to deal with missing values. If students enrolled in a course but did not take the exam, a 6.0 was imputed; if they were not enrolled at all, a 7.0 was imputed. This means that not enrolling (7.0) is penalized more than enrolling but not taking the exam (6.0).

Data representation. Each student is represented by a vector of grades. It is possible for a student to, for example, enroll in a course in the 1st semester and not take the exam, then enroll and fail the exam in the next semester and enroll again and pass the exam in the following semester. In this case, a student has three different records for the same course in three different semesters. In our opinion, not only the final grade with which a course was passed is relevant, so we include the entire history of a student's academic performance in the vector. Table 3 shows the vector representation of six students for their three first semesters of study. Note that the courses where all students have the value 7.0 are omitted. Students 0, 3, and 5 enrolled in the course M03 without taking the exam in semester 1 (value 6.0), students 0 and 3 did the same in semester 2 but did not enroll in semester 3 (value 7.0), while student 5 did the opposite; students 1, 2, and 4 passed M03 in semester 1.

Table 3: Example of a course recommendation for one student with five neighbors for the 3rd semester. The columns show the semesters (1, 2, 3) and the courses the students were enrolled in, e.g. M01, M02. Row 0 represents the student who receives a recommendation, and rows 1 to 5 represent the student's five nearest neighbors. The recommended courses are highlighted in blue. The cells show their grades; 6.0 and 7.0, colored in gray, are imputed for missing values. The actual grades of student 0 in semester 3 are given for comparison and highlighted in italic.

S	1					2							3								
	M01	M02	M03	M04	M05	M03	M08	M09	M10	M11	M12	M13	M03	M10	M11	M14	M15	M16	M17	M18	M19
0	1.3	2.7	6.0	2.3	2.0	6.0	1.3	2.3	7.0	2.7	2.7	2.7	*7.0*	*2.3*	*7.0*	*3.0*	*3.0*	*2.3*	*2.7*	*2.0*	*7.0*
1	1.7	2.7	2.0	2.7	2.3	7.0	2.0	2.7	6.0	4.0	2.3	2.3	7.0	7.0	7.0	1.3	1.3	1.7	3.0	1.7	4.0
2	2.0	1.7	2.3	2.3	1.7	7.0	1.3	2.3	7.0	3.0	2.3	5.0	7.0	7.0	7.0	2.0	2.0	7.0	2.0	2.7	1.3
3	2.3	2.0	6.0	2.7	2.0	6.0	1.3	1.7	6.0	6.0	2.0	2.0	7.0	7.0	6.0	2.7	1.7	1.7	3.0	2.0	2.7
4	2.3	2.3	3.0	2.3	1.0	7.0	2.7	2.0	6.0	1.3	2.3	2.3	7.0	7.0	7.0	1.7	1.7	2.0	1.0	1.3	3.7
5	1.3	3.0	6.0	2.3	3.0	7.0	1.7	6.0	6.0	2.3	2.7	2.3	6.0	7.0	7.0	3.0	5.0	2.7	6.0	5.0	7.0

4. METHODOLOGY

In this section, we present first the course recommender system. Then we explain the two-step dropout prediction and how we optimized the models. Finally, we describe the evaluation of the prediction models for RQ1 and the course recommendations for RQ2 and RQ3. In our case, since many students drop out after the first or second semester, we consider the recommendations and dropout predictions for the second and third semesters. For each research question, we look at subgroups by program, semester, and student status.

4.1 Course Recommendations

The course recommender system is based on a KNN classifier: given a student represented by a vector of grades at the end of semester t, the majority votes of his/her neighbors classify a course as "passed" and accordingly recommended for the following semester $t+1$. KNN has the advantage that the neighbors can be calculated only once and on their basis, the classification can be made for all courses. Since we considered all courses passed by any student in semester $t+1$, we got two sets: "recommended" and "not recommended". Given the possibility to recommend a course that the student being observed has already passed, we removed this course from the recommendation if necessary. We recommended courses for all 1,366 students to have the largest possible database to evaluate the recommendations.

Parameters. To avoid a tie in majority voting, we used only uneven k and tested our approach with k from 1 to 25. Additionally, we selected the Euclidean Distance as distance metric for calculating the distances between the students.

Risk reducing approach and baseline. To ensure that courses are recommended that reduce dropout risk, we included in our approach only neighbors who graduated from the program. As a baseline for comparison, we used also all neighbors, which means including neighbors who dropped out, to generate course recommendations. We expected that the recommendations differ depending on the neighbor type and that the recommendations based on graduated students, but not necessarily the recommendations generated with all students, reduce the risk of dropping out. In the following, we distinguish the two neighbor types with AN (all neighbors) and GN (neighbors who graduated).

Example. Table 3 shows the data used to calculate the neighbors and to recommend courses to student 0 for the 3rd semester. The actual grades — or imputed values 6.0 and 7.0 if grades were missed — for relevant courses (M01 to M19) are shown for each semester. Semesters 1 and 2 are the previous semester on which the distance calculation is based. Semester 3 covers the course recommendations. A course is passed if the grade lies between 1.0 and 4.0. M10 was not recommended to student 0 in this example but student 0 passed it in semester 3, M19 was recommended because four of five neighbors passed it but student 0 did not enroll in it.

4.2 Dropout Risk Prediction

A dropout prediction was performed using the following two steps: 1) A model was trained to predict the two classes: dropout (D) or graduate (G) based on actual enrollment and exam information; 2) The model from step 1 was used again to predict dropout or graduation after the calculated recommendations replaced the actual enrollment and exam information. We call "dropout risk" the proportion of students in the test set who are predicted to drop out in this prediction task. To determine whether or not the recommended courses help to lower the dropout risk, we compare the predicted dropout risk from step 1 (P_1) with the predicted dropout risk from step 2 (P_2). The goal is for P_2 to be less than P_1.

4.2.1 Step 1

Feature set. As investigated by Manrique et al. [11], there are several ways to select a feature set for dropout prediction and no way works better than the others in all contexts. Because we want to measure the impact of our recommendations on dropout prediction, we use the courses taken by students as features; the values of the features are the grades.

Model training. To detect a change in the dropout risk, the models should be as accurate as possible which we aimed to achieve through two approaches: A) train different types of algorithms, and B) use different approaches for optimization. For all cases, the datasets were sorted by students' study start and then split into 80% training data and 20% test data, so that risk prediction is done for students who started their studies last. As can be seen in Table 1, the proportion of dropouts is higher in the test set than in the training set because it usually takes six semesters to know whether a student will graduate whereas many students drop out of their studies much earlier. We trained models for each program (AR, CM, PT) and semesters $t = 2$ and $t = 3$ with actual grades and used the best models to evaluate a change in dropout risk in step 2).

Algorithms. We trained the following algorithms in Python using scikit-learn: decision tree (DT), lasso (L, penalty=l1, solver=liblinear), logistic regression (LR, penalty=none, solver=lbfgs), k-nearest neighbors (KNN), random forest (RF), support vector machine with different kernels (SV: rbf, LSV: linear, PSV: poly).

Optimization. Using our experience in [20], we simply use the scikit-learn default hyperparameter settings, except the settings to obtain a certain algorithm as mentioned above, in combination with the following list of algorithm-independent parameters. i) Feature selection by cut-off (CO): we removed courses with too few grades and tried 1 and 5 as a minimum number of grades to retain a course; a value that is too high may result in the removal of recommended courses and thus would not be included in the dropout prediction. ii) Training data balancing (BAL): we used two common techniques: Synthetic Minority Oversampling Technique (SMOTE) [4] and RandomOverSampler (ROS). Both implementations are from imbalanced-learn, a Python library. iii) Decision threshold moving (DTM): Usually, a classifier decides for the positive class at a probability greater or equal to 0.5, but in case of imbalanced data, it may be helpful to adjust this threshold, so we checked additionally to 0.5 values between 0.3 and 0.6 in 0.05 steps. Lower and higher values did not lead to better results.

Evaluating the model performance. To emphasize that both correct dropouts and correct graduates are important for dropout risk prediction, we evaluated the models based on the test data using the Balanced Accuracy metric (BACC), defined as the mean of the recall for class 1 (dropout), also known as true positive rate, and recall for class 0 (graduate), also known as true negative rate: $BACC = (TP/P + TN/N)/2$.

4.2.2 Step 2

In the second step, we used the best model by BACC from step 1 for each program and the semesters $t = 2$ and $t = 3$ to predict dropout. The dropout prediction for $t = 2$ used the actual grades of the 1st semester and the recommendations for the 2nd semester, and the prediction for $t = 3$ used the actual grades of the 1st and 2nd semesters and the recommendations for the 3rd semester. For the recommendations, we assumed that the student can pass the recommended courses. For student 0 in Table 3, courses from M14 to M19 are recommended and we assume that s/he will pass all these courses in semester 3. If we had an actual grade in the records for that student and a recommended course, we used it. If not, we predicted a grade by imputing the average of two medians: the median of all the grades that we know about from the student and the median of the historical grades for that course. This imputation rests on the strong assumption that underpins our recommendations: the majority vote of the k nearest neighbors yields a set of courses that a student can pass. We evaluated this grade prediction using the known actual grades and obtained a Root Mean Square Error (RMSE) of 0.634, which is comparable with RMSE scores from 0.63 to 0.73 to other studies in that field [6, 16]. Consider again student 0 in Table 3. In addition to the courses from semesters 1 and 2, M10 and M14 to M18 from the third semester were used for prediction in step 1, and M14 to M19 from the third semester were used in step 2 with a predicted grade for M19.

Table 4: Structure of the confusion matrix for recommendation evaluation for one student.

	Predicted positive	**Predicted negative**	**Totals**
Actual positive	Passed and recommended True positive TP	Passed but not recommended False negative FN	**Passed P**
Actual negative	Not passed but recommended False positive FP	Not passed and not recommended True negative TN	**Not passed**
Totals	**Recommended**	**Not recommended**	**All courses**

4.3 Evaluation

4.3.1 RQ1 Evaluation

To answer the question ”Do the recommendations lower the risk of dropping out?” in section 5.1, we compare the dropout risk, i.e. the proportion of students who are predicted to drop out, based on the predictions from step 2 (P_2) with those from step 1 (P_1). We also distinguish the neighbor types for step 2: P_{2AN} corresponds to the step 2 dropout prediction using the courses recommendations based on all neighbors (baseline) while P_{2GN} uses the recommendations based on graduate neighbors.

4.3.2 RQ2 Evaluation

Since the course recommendations are for each course a binary classification problem, we employ a confusion matrix for each student (Table 4) to answer the question ”How large is the intersection between the set of courses recommended and the set of courses a student has passed?” in section 5.2. We evaluate the recommendation for semester $t + 1$ for each student as follows: a course recommended and actually passed is a true positive (TP), a course recommended and actually not passed is a false positive (FP), a course not recommended but passed is a false negative (FN), and a course not recommended and not passed is a true negative (TN).

To evaluate a set of recommended courses, it’s important to measure both recall (whether passed courses include recommended courses) and precision (whether recommended courses include passed courses). We chose the F_1 score to evaluate courses’ intersections since the F_1 score ignores TN, which is in our context always a high value and thus does not serve our needs. The score ranges from 0 to 1 with 1 indicating perfect classification (recall=1 and precision=1) and 0 indicating perfect misclassification (recall=0 or precision=0). The calculation is as follows: $F_1 = 2 \cdot TP/(2 \cdot TP + FP + FN)$.

Further, we provide the recall which is in our case TP/P and equivalent to recall@ns, the percentage of recommended courses based on the number of courses taken by student s to enable comparison with similar work [10, 17]. Recall@n would fix the number of recommended courses at n [6, 14] and is not applicable in our case since we do not rank the recommendations and may also recommend less than n courses. Looking at the recommendations for student 0 in Table 3, the courses M14 to M18 are TP, M10 is FN, M19 is FP, and all the other 29 — here not shown — courses are TN. $F_1 = 2 \cdot 5/(2 \cdot 5 + 1 + 1) = 0.8\bar{3}$. $Recall = 5/6 = 0.8\bar{3}$. We aggregate the results as mean F_1 for both neighbor types and mean recall for neighbor type GN of all students grouped by student status, type of neighbors, program, and semester to compare the scores of the subgroups.

4.3.3 RQ3 Evaluation

To answer the question "a) How many courses are recommended and b) does this number differ from the number of courses passed and enrolled in by students?" in section 5.3, we look first at the number of courses recommended for semester $t + 1$. Using a horizontal barplot, we visualize the distribution of students by the number of recommended courses. To analyze why some students get no or only a few recommendations, we describe the relationship between the number of recommended courses and the distance of students to their neighbors. Using a scatterplot, we visualize the mean distance of a student to its neighbors in relation to the number of recommended courses. Second, we calculate the median difference between the number of courses recommended and courses passed, and the median difference between the number of courses recommended and courses enrolled. This may yield a difference in the number of courses students pass or enroll in than recommended by the system, depending on the subgroup.

5. RESULTS AND DISCUSSION

In this section, we first present the dropout prediction models and the changes in dropout risk based on the two-step prediction (RQ1). This includes identifying an appropriate value for k, the number of neighbors, that we use for the in-depth analysis of the course recommendations regarding the intersection (RQ2), and the number of courses (RQ3).

5.1 Dropout Risk

5.1.1 Dropout Prediction Models

Step 1 prediction. We selected the models — trained with actual exam and enrollment data — with the highest BACC for each program and semester (Table 5). They differ regarding their algorithm-independent parameters. We obtain P_1 as the step 1 dropout risk, i.e., the proportion of students from the test set predicted to drop out, that we compare later with the step 2 dropout risk P_2.

Example CM2. The support vector classifier (column C) achieved the best BACC when removing all courses that do not have at least one grade (column CO) resulting in 36 courses or features (column F); the decision threshold (column DTM) is 0.3, which means that students are predicted to drop out already at a 30% probability; the training set was not balanced (column BAL). Compared to the actual risk in the test data (0.632, Table 1 row CM > test), the predicted risk is lower (0.557).

Table 5: Best step 1 dropout prediction models for programs and semester (PS) regarding balanced accuracy (BACC) including their corresponding recall (REC), the classifier used (C), the number of used features (F), optimized parameters (CO, DTM, BAL), and the proportion of students of the test set who are predicted to drop out ($\mathbf{P_1}$).

PS	C	F	CO	DTM	BAL	$\mathbf{P_1}$	BACC	REC
AR2	RF	38	0	0.35	SMOTE	0.353	0.866	0.814
AR3	RF	32	4	0.45	ROS	0.336	0.935	0.884
CM2	SV	36	1	0.30	None	0.557	0.920	0.866
CM3	RF	74	0	0.45	SMOTE	0.566	0.927	0.881
PT2	LSV	16	3	0.30	SMOTE	0.358	0.913	0.857
PT3	LSV	47	3	0.30	SMOTE	0.396	0.882	0.857

The best models have been obtained when the training data is balanced except for program CM and semester 2. The predicted dropout risk P_1 is lower in all cases than the actual dropout risk, see column Risk for the test set in 1, as we have observed for CM2, except for PT3 where it is equal. This means that our models tend to be optimistic and predict as graduate some students who dropped out.

5.1.2 Changes in Dropout Risk

Using the best models shown in Table 5, we performed the step 2 prediction using the recommendations.

Selecting an appropriate value for *k*. The set of recommended courses is critical for the step 2 prediction and depends on the number of neighbors *k*. Unfortunately, our research has shown that there is no value of k that generates an optimal set of courses for all three study programs and semesters and the two kinds of students: those who dropped out and those who graduated. Two values, $k = 3$ and $k = 5$, emerge as optimal or near-optimal and as never bad. The neighbors provide students with examples of how fellow students have enrolled and passed courses in their studies; this is one support that our students are looking for when they enroll [19]. Acknowledging this wish, matching the number of similar people used in the interviews by Du et al. [5], and in order to provide students with a variety of paths through their studies that are close to their own path, we choose $k = 5$ for further analysis in this work.

Step 2 prediction. Table 6 shows three proportions of students who are predicted as dropouts using the recommendations of five neighbors: P_1 from step 1, P_{2GN} based on neighbors who graduated, and P_{2AN} based on all neighbors. We distinguish the predicted dropout risk by student status, D or G, for a better overview of how the models perform.

Example CM2. Considering students who actually dropped out (D), 86.6% are predicted to drop out in step 1, 77.6% in step 2 using recommendations calculated with all neighbors

Table 6: Mean predicted dropout risk in step 1 ($\mathbf{P_1}$) and based on five neighbors and both neighbor types (AN, GN) in step 2 ($\mathbf{P_2}$) by student status (D, G), program and semester (PS). $\mathbf{P_{2GN}}$-$\mathbf{P_1}$ gives the corresponding change.

ST	PS	$\mathbf{P_1}$	$\mathbf{P_{2AN}}$	$\mathbf{P_{2GN}}$	$\mathbf{P_{2GN}}$-$\mathbf{P_1}$
D	**AR2**	0.814	0.674	0.558	-0.256
	AR3	0.884	0.721	0.279	-0.605
	CM2	0.866	0.776	0.716	-0.149
	CM3	0.881	0.821	0.716	-0.164
	PT2	0.857	0.619	0.619	-0.238
	PT3	0.857	0.905	0.810	-0.048
G	**AR2**	0.082	0.014	0.027	-0.055
	AR3	0.014	0.041	0.000	-0.014
	CM2	0.026	0.051	0.051	0.026
	CM3	0.026	0.051	0.026	0.000
	PT2	0.031	0.406	0.312	0.281
	PT3	0.094	0.250	0.188	0.094

(AN), and 71.6% using recommendations calculated with neighbors who graduated (GN). Looking at students who actually graduated (G), 2.6% are predicted to drop out in step 1, 5.1% in step 2 using recommendations calculated with all neighbors, and also 5.1% using recommendations calculated only with students who graduated. Thus, if we use the course recommendations and assume that these exact courses are passed, the risk decreases by 14.9% for actual dropouts and increases by 2.6% for actual graduates. Based on the size of the test dataset (Table 1), this means in absolute numbers: of 67 dropouts, 10 more students are predicted to graduate and of the 39 graduates, one more student is predicted to drop out compared to the step 1 prediction.

5.1.3 RQ1 Findings and Discussion

The question "Do the recommendations lower the risk of dropping out?" can be answered with yes, our approach lowers the dropout risk in most cases and we explore the risk reduction scores from different perspectives more precisely:

Graduates and dropouts. As we analyze Table 6, we expect the values in column P_{2GN} to be equal to or smaller than those in column P_1, and this holds true for students with status D, who are the primary focus of our recommendations. Additionally, for the AR program, we observe the same pattern for students with status G. However, for the program CM semester 2 and the program PT, the values in column P_{2GN} are higher than those in column P_1 , specifically for the graduated students. A glance at Table 1 reveals that the number of students with status G is small in the test set of CM2, while the program PT has a smaller number of students overall than the other two programs. This could explain these somewhat negative results, particularly for the PT program.

AN-based and GN-based recommendations. Comparing column P_{2AN} of Table 6 with column P_1, one notices that the values are everywhere smaller or equal in column P_{2AN} for the students with status D, except for PT3. This is less true for the students with status G. Comparing column P_{2AN} with column P_{2GN}, one notices that the values in column P_{2GN} are everywhere smaller or equal, except for the students with status G in AR2. These results indicate that calculating the recommendations by choosing the neighbors among all students could already be helpful. They also confirm that choosing neighbors among the students who graduated gives better results.

2nd and 3rd semester. Looking at the column $P_{2GN}-P_1$, we expect all values for the students with status G to be small, as not many students who graduated are predicted to drop out; one notices the small value -0.048 in PT3 for the students with status D. We conjecture that this is due to the high number of elective courses proposed in semester 3 of this study program. As students can freely choose five courses from six among a list of about 25 courses, it is more difficult for the algorithm to calculate accurate recommendations.

Overall, the results show that students who dropped out will benefit from enrolling and passing the courses recommended to them, above all when the recommendations are calculated with neighbors who have graduated. The assumption that students will pass the courses recommended to them sounds strong. However, as we shall see in section 5.3, the number of recommended courses is on average one course less than the number of enrolled courses. Focusing on fewer courses, as the recommendations suggest to them, might be helpful.

The findings indicate that the utilization of machine learning algorithms for assessment purposes may be constrained in scenarios where the student population is limited, particularly in the context of degree programs CM and PT with a small number of students possessing status G. The outcomes generated may not be reliable due to the small sample size. Additionally, the study reveals a limitation in recommendations based on nearest neighbors when the degree program is configured with a substantial number of elective courses, such as in the third semester of program PT. Therefore, relying on such recommendations may not be suitable in this particular scenario.

5.2 Courses' Intersection

We evaluate how the set of recommended courses calculated with five neighbors intersects with the set of courses students have passed using the means of the individual F_1 scores and recall (Table 7). To better distinguish for which student groups the recommendations better align with actual courses passed, the results are grouped by program and semester (PS), student status (D, G), and type of neighbors (AN, GN). Note that recall is shown when recommendations are calculated with neighbors from the set GN.

Example CM2. The F_1 score for students who actually dropped out (D) is 0.328 for recommendations based on all neighbors (AN) and 0.397 for recommendations based only on neighbors who graduated (GN). Looking at students who graduated (G), the F_1 score is much higher, 0.824 for rec-

Table 7: Mean F_1 score for neighbor types (AN, GN) and mean recall for neighbor type GN by student status (D, G), program, and semester (PS).

	$\mathbf{F_{1AN}}$		$\mathbf{F_{1GN}}$		$\mathbf{Recall_{GN}}$	
PS	**D**	**G**	**D**	**G**	**D**	**G**
AR2	0.481	0.854	0.521	0.871	0.649	0.925
AR3	0.279	0.817	0.305	0.842	0.417	0.875
CM2	0.328	0.824	0.397	0.851	0.498	0.895
CM3	0.130	0.711	0.159	0.755	0.187	0.788
PT2	0.511	0.837	0.528	0.828	0.651	0.844
PT3	0.112	0.335	0.156	0.356	0.140	0.284

ommendations based on all neighbors (AN) and 0.851 for recommendations based on neighbors who graduated (GN). Recall is 0.498 for students with status D and 0.895, again much higher, for students with status G.

5.2.1 RQ2 Findings and Discussion

We look at the question ”How large is the intersection between the set of courses recommended and the set of courses a student has passed?” from different perspectives.

Graduates and dropouts. The recommendations should show another, more promising way of studying to students who are struggling while they should not disturb students who are doing well. Thus, we expect the F_1 score and recall to be much higher for students with status G than for students with status D. We consider only the two columns GN on the right of Table 7 in the remainder of this section, namely recommendations calculated using neighbors who graduated as they gave the best F_1 results, which means that overall, graduate neighbors recommend better the courses that the students have actually passed. The column F_{1AN} is shown for the seek of completeness. As expected, the mean F_{1GN} score and recall are always higher for students with status G than for students with status D. F_{1GN} is higher than 82% in four cases and 75.5% in one case. Recall is always higher than 78%. This means that the recommended courses reflect quite well how these students study. An exception is program PT and semester 3. This might be due to the high number of elective courses offered by that program in semester 3. Of the 26 courses recommended to at least one student and also used in dropout prediction, only one is mandatory; the other 25 are electives. For students with status D, the mean F_{1GN} score tends to be low, around 52% in two cases and below 40% in the other cases.

2nd and 3rd semester. The mean F_1 score and the mean recall are higher in all cases for the 2nd semester than for the 3rd semester. The higher the semesters, the more the courses students pass drift apart. On the one hand, this makes it more difficult to find close neighbors, and on the other hand, it makes the recommendation itself more difficult: the neighbors sometimes disagree and have passed too many different courses, which means that no majority can be found for many courses and these courses are not recommended. This is particularly true for PT3 because of the high number of elective courses, as already mentioned.

Overall, the results indicate that the recommended courses match quite well the courses passed by students who graduated and show another way of studying to students who dropped out. The results also confirm a limitation of the proposed recommendations when the study degree program foresees many elective courses in a semester. For comparison with related work, we provide the mean F_{1GN} score for all students across programs and semesters with a value of 0.646 and the mean $Recall_{GN}$ with a value of 0.689. Depending on the semester, the scores of Ma et al. range from 0.431 to 0.472 [10] and Polyzou et al. obtain an overall mean score of 0.466 [17].

5.3 Number of Recommended Courses

We answer the questions ”a) How many courses are recommended and b) does this number differ from the number of courses passed and enrolled in by students?” in two parts.

5.3.1 Number of Recommended Courses

Figure 1 contrasts the number of recommended courses based on all neighbors and students who graduated. As already written, the recommendations are calculated with five neighbors. Their number varies between 0 and 7 in both cases. The charts show for each number the respective percentage of students grouped by status (D, G), program (AR, CM, PT), and semester (2, 3). When comparing the top and bottom charts of Figure 1, it is clear that recommendations calculated with all neighbors result in an empty set, i.e., 0 courses recommended, more frequently than recommendations calculated only with students who graduated. This confirms that the recommendations calculated only with neighbors who graduated give better results. Therefore, and as before, we consider only the recommendations calculated with neighbors who graduated in the remaining of this section.

Example CM2. In the upper half of the GN chart (bottom of Figure 2), we begin with row G-CM-2. According to the handbook, more than half of the students who graduated get six courses recommended, about 20% get five courses recommended, and the remaining students get four, three, or two courses recommended; a few students get one; no student gets an empty set. Row D-CM-2 is now under consideration. The picture looks different. About 50% of the students are recommended four or three courses, over 30% are recommended six or five courses, and the remaining students are recommended two or one course; no student is recommended an empty set.

Further investigation of the small number of courses recommended. Since some students do not receive any recommendations, see for example the rows CM3 and AR3, we examined the number of recommended courses as a function of the distance between students and their neighbors. Figure 2 shows for program CM a scatter-plot of the mean distance of the students from their neighbors (y-axis) by the number of recommended courses (x-axis) distinguishing status D and status G; semester 2 is on the left, semester 3 on the right. We can observe that when neighbors are farther

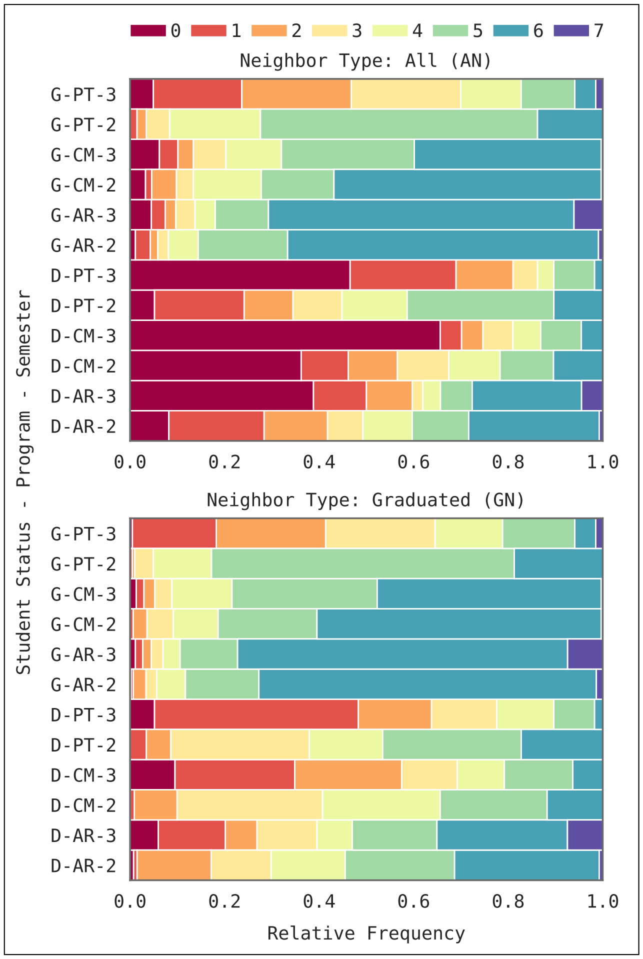


Figure 1: Distribution of students by the number of recommended courses (0 to 7), student status (D, G), program (AR, CM, PT), and semester (2, 3); top: neighbor type AN, bottom: neighbor type GN.

away, fewer courses are recommended. The trend is similar for the status dropout, though less drastic, and for the 3rd semester; it is also similar for the two other programs, not represented here. As an example, students with good grades but enrolling in only part of the courses in semesters 1 and 2, might be far from their nearest neighbors because of the imputed value of 7.0 for the courses not enrolled in.

RQ3a) Findings and Discussion. The percentage of students who receive no recommendation or only one course recommended is much smaller when the recommendations are calculated with neighbors who graduated than with all neighbors. This is especially noticeable for students who dropped out. This finding confirms again the superiority of calculating the recommendations with GN. For graduates in AR, CM, and PT in semester 2, the number of recommended courses is for the majority of the students close to the number planned in the curriculum, i.e., five or six courses. Again, PT3 differs. As is visible in the evaluation of the intersection in section 5.2, there is less agreement about the courses among the neighbors, which can be explained by a large number of elective courses in semester 3. This leads to smaller set sizes regarding course recommendations. Our results show also that students who are very different from their neighbors, especially those with status G, are likely to get few recommendations.

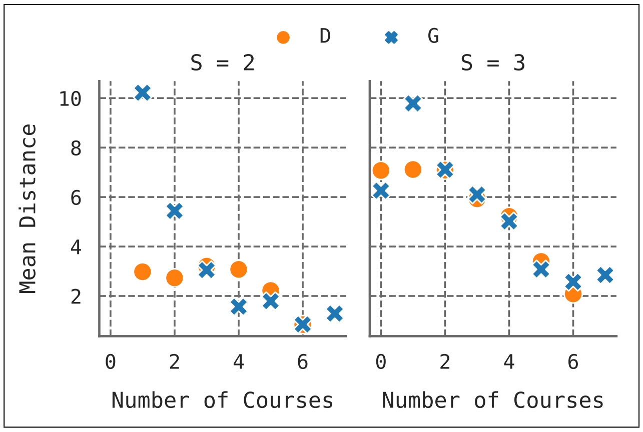


Figure 2: Mean Distance from neighbors by number of recommended courses for program CM; left: semester 2, right: semester 3. Markers and colors correspond to student status D and G.

5.3.2 *Numbers: Recommended, Enrolled, and Passed*

Table 8 provides the difference between the median number of courses recommended and the median number of courses enrolled (R - E) or passed (R - P). To better distinguish for which student groups the recommendations are closer to the actual numbers, the results are grouped by status (D, G), neighbors type (AN, GN), program, and semester PS. Note that the results with two kinds of neighbors, AN and GN, are shown for the seek of completeness. We only discuss the results calculated with neighbors who graduated, GN, as these results are better.

Example CM2. We consider first students who dropped out (D). The column (R - E) > GN has the value -1.0, which means that the number of recommended courses is on average 1 less than the number of courses the students enroll in. Comparing the number of recommended courses with the number of those passed (R - P) > GN, we see a value of 2.0, meaning that the number of recommended courses is on average 2 more than the number of courses the students pass. Considering students who graduated, we see no difference in the number of courses recommended, enrolled in, and passed on average: all values are 0.

RQ3b) Findings and discussion. On the one hand, the recommender system suggests to students who dropped out to focus on fewer courses, the column (R - E) > GN has everywhere negative values, i.e., enroll in fewer courses with the expectation that they can pass more courses instead, the column (R - P) > GN has everywhere positive values, except in PT3. On the other hand, nothing changes on average for graduates: there is no difference, except for PT3. The problem with PT3 is the lower number of recommended courses in general, as also visible in Figure 1, which can be explained by a large number of elective courses, as already written.

Table 8: **Median difference between the number of courses recommended and the number of courses enrolled in (R-E) and the number of courses passed (R-P) by student status (D, G), neighbor type (AN, GN), program, and semester (PS).**

	D				G			
	R - E		R - P		R - E		R - P	
PS	AN	GN	AN	GN	AN	GN	AN	GN
AR2	-2.0	-1.0	0.5	1.0	0.0	0.0	0.0	0.0
AR3	-3.0	-1.0	0.0	2.0	0.0	0.0	0.0	0.0
CM2	-3.0	-1.0	0.0	2.0	0.0	0.0	0.0	0.0
CM3	-4.0	-2.0	0.0	1.0	-1.0	0.0	0.0	0.0
PT2	-2.0	-1.0	1.0	1.0	0.0	0.0	0.0	0.0
PT3	-5.0	-4.0	-0.5	0.0	-4.0	-3.0	-3.0	-3.0

6. PRELIMINARY USER EVALUATION

The approach has been evaluated with 12 students of the study program CM as part of an assessment in the elective course "machine learning". Students were in their 4th or 5th semester, and all performed well in their first two semesters. Beforehand, students had the possibility to hand in their records anonymously and have recommendations calculated for semesters 2 and 3. Three students made use of this possibility. The recommendations were identical to the courses that they actually passed in three cases (F_1=100%). The three other cases had an F_1 score of 90.1%, 86,1%, and 0%, respectively. The last case refers to a student with relatively good grades who enrolled in three courses only in semester 2 resulting in an average distance of 13 from the neighbors. Overall, these results confirm our assumption that, for students with good academic performance, the recommendations should closely match the courses that they pass.

The evaluation mainly consisted of a semi-guided group discussion concerning the recommendations. We report here the answers and discussion to two questions: 1. Are the recommendations understandable? 2. Would you use such a recommender system? All groups answered the first question with yes but also gave ideas for improvement. For example, they considered three to five neighbors to be the most useful, as this is the quickest and clearest way to grasp how the recommendations come about. This fits very well with the dimensions of interpretability that Guidotti et al. give [8], namely "time limitation" but also "nature of user expertise". Six students answered the second question with yes, four with no, and two were undecided. One main reason not to use such a system was the following: seeing the grades of others can be stressful: will I perform as well as the given examples? Interestingly, an undecided student said that it might be encouraging to see that other fellow students did not always get good grades but were able to graduate. These utterances are similar to the findings in [3]. More evaluations, particularly with students who are unfamiliar with machine learning, are required to study the interpretability and related trust in the recommendations.

7. CONCLUSION AND FUTURE WORK

This paper presents a comprehensive evaluation of a novel course recommender system designed to primarily support students who face difficulties in their initial semesters and are at risk of dropping out. The evaluation utilizes data from three distinct study programs that vary in terms of their subject matter, student population, and program structure, including a program with a high number of elective courses in the third semester.

The evaluation of the first research question indicates that, overall, the recommendations lead to a reduction in the dropout rate, particularly for the targeted at-risk students who dropped out. However, the results are less conclusive for students who graduated, which may be due to the limited data available in the test set.

The evaluation of the second research question reveals that the recommended courses generally align with the courses that graduated students passed, except for the 3rd semester of program PT, which contains many elective courses. This is not the case for students who dropped out, as the recommendations suggest a different approach to their studies.

The evaluation of the third research question demonstrates that the number of recommended courses is close to the number of courses planned in the curriculum for graduating students, except for the aforementioned 3rd semester of program PT. However, for students who dropped out, the number of recommended courses is generally lower than the number of courses they enrolled in.

Overall, the evaluations have revealed two main limitations of our recommender system. Primarily, it is better suited for curricula consisting mostly of mandatory courses that all students must pass, as is often the case in the first two semesters of a program. Additionally, it recommends very few courses for students with distant neighbors, and therefore, a different approach to handling passed courses in the recommender system should be explored. However, it does allow for presenting the paths of five neighbors as an impulse.

Summing up, the paths followed by students who graduated are helpful to other students, especially those who struggle. It is worth noting that our approach to course recommendation is generalizable even if enrollment data is not stored, as is the case in some institutions. Except for comparing the number of recommended courses to the number of enrolled courses, the evaluation remains the same.

A preliminary evaluation with students indicates that the recommendations are understandable. Further research with 2nd or 3rd semester students is planned to determine how ready and willing they are to use such recommendations as well as the advantages of using sets instead of rank lists. In addition, it is necessary to evaluate whether students understand the recommendations and what additional support they need to pass all recommended courses, aside from taking fewer and different courses than they might think. As stated in the German context [7], a combination of well-orchestrated interventions usually leads to academic success.

8. REFERENCES

[1] L. Aulck, D. Nambi, N. Velagapudi, J. Blumenstock, and J. West. Mining University Registrar Records to Predict First-Year Undergraduate Attrition. In *12th International Conference on Educational Data Mining (EDM)*, pages 9–18, Montreal, Canada, 2019. International Educational Data Mining Society. https://eric.ed.gov/?id=ED599235.

[2] J. Berens, K. Schneider, S. Gortz, S. Oster, and J. Burghoff. Early Detection of Students at Risk - Predicting Student Dropouts Using Administrative Student Data from German Universities and Machine Learning Methods. *Journal of Educational Data Mining*, 11(3):1–41, 2019. https://zenodo.org/record/3594771.

[3] A. Brun, B. Gras, and A. Merceron. Building Confidence in Learning Analytics Solutions: Two Complementary Pilot Studies. In D. Ifenthaler and D. Gibson, editors, *Adoption of Data Analytics in Higher Education Learning and Teaching*, Advances in Analytics for Learning and Teaching, pages 285–303. Springer International Publishing, Cham, 2020. https://doi.org/10.1007/978-3-030-47392-1 _15.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. http://arxiv.org/abs/1106.1813.

[5] F. Du, C. Plaisant, N. Spring, and B. Shneiderman. Finding Similar People to Guide Life Choices: Challenge, Design, and Evaluation. In *Proceedings of the 2017 Conference on Human Factors in Computing Systems (CHI)*, pages 5498–5544, New York, NY, USA, 2017. Association for Computing Machinery. https://doi.org/10.1145/3025453.3025777.

[6] A. Elbadrawy and G. Karypis. Domain-Aware Grade Prediction and Top-n Course Recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys)*, pages 183–190, New York, NY, USA, 2016. Association for Computing Machinery. https://doi.org/10.1145/2959100.2959133.

[7] S. Falk, M. Tretter, and T. Vrdoljak. Angebote an Hochschulen zur Steigerung des Studienerfolgs: Ziele, Adressaten und Best Practice. *IHF kompakt*, (March 2018), 2018. https://www.ihf.bayern.de/publikationen/ihf-kompakt/detail/angebote-an-hochschulen-zur-steigerung-des-studienerfolgs-ziele-adressaten-und-best-practice.

[8] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A Survey of Methods for Explaining Black Box Models. *ACM Computing Surveys*, 51(5):1–42, 2018. https://doi.org/10.1145/3236009.

[9] L. Kemper, G. Vorhoff, and B. U. Wigger. Predicting Student Dropout: A Machine Learning Approach. *European Journal of Higher Education*, 10(1):28–47, 2020. https://doi.org/10.1080/21568235.2020.1718520.

[10] B. Ma, Y. Taniguchi, and S. Konomi. Course Recommendation for University Environments. In *Proceedings of the 13th International Conference on Educational Data Mining (EDM)*, pages 460–466, Online, 2020. International Educational Data Mining Society. https://eric.ed.gov/?id=ED607802.

[11] R. Manrique, B. P. Nunes, O. Marino, M. A. Casanova, and T. Nurmikko-Fuller. An Analysis of Student Representation, Representative Features and Classification Algorithms to Predict Degree Dropout. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge (LAK)*, pages 401–410, New York, NY, USA, 2019. Association for Computing Machinery. https://doi.org/10.1145/3303772.3303800.

[12] S. Morsy and G. Karypis. Will This Course Increase or Decrease Your GPA? Towards Grade-Aware Course Recommendation. *Journal of Educational Data Mining*, 11(2):20–46, 2019. https://eric.ed.gov/?id=EJ1230292.

[13] D. Novoseltseva, K. Wagner, A. Merceron, P. Sauer, N. Jessel, and F. Sedes. Investigating the Impact of Outliers on Dropout Prediction in Higher Education. In *Proceedings of DELFI Workshops 2021*, pages 120–129, Online, 2021. Gesellschaft für Informatik e.V.z. https://nbn-resolving.org/urn:nbn:de:hbz:1393-opus4-7338.

[14] Z. A. Pardos, Z. Fan, and W. Jiang. Connectionist recommendation in the wild: on the utility and scrutability of neural networks for personalized course guidance. *User Modeling and User-Adapted Interaction*, 29(2):487–525, 2019. https://doi.org/10.1007/s11257-019-09218-7.

[15] Z. A. Pardos and W. Jiang. Designing for serendipity in a university course recommendation system. In *Proceedings of the 10th International Conference on Learning Analytics & Knowledge (LAK)*, pages 350–359, New York, NY, USA, 2020. Association for Computing Machinery. https://doi.org/10.1145/3375462.3375524.

[16] A. Polyzou and G. Karypis. Grade Prediction with Course and Student Specific Models. In *Advances in Knowledge Discovery and Data Mining. 20th Pacific-Asia Conference (PAKDD)*, pages 89–101, Auckland, New Zealand, 2016. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-31753-3 _8.

[17] A. Polyzou, A. N. Nikolakopoulos, and G. Karypis. Scholars Walk: A Markov Chain Framework for Course Recommendation. In *Proceedings of the 12th International Conference on Educational Data Mining (EDM)*, pages 396–401, Montreal, Canada, 2019. International Educational Data Mining Society. https://eric.ed.gov/?id=ED599254.

[18] M. C. Urdaneta-Ponte, A. Mendez-Zorrilla, and I. Oleagordia-Ruiz. Recommendation Systems for Education: Systematic Review. *Electronics*, 10(14):1611, 2021. https://doi.org/10.3390/electronics10141611.

[19] K. Wagner, I. Hilliger, A. Merceron, and P. Sauer. Eliciting Students' Needs and Concerns about a Novel Course Enrollment Support System. In *Companion Proceedings of the 11th International Conference on Learning Analytics & Knowledge (LAK)*, pages 294–304, Online, 2021. https://www.solaresearch.org/core/lak21-companion-

proceedings/.

[20] K. Wagner, A. Merceron, P. Sauer, and N. Pinkwart. Personalized and Explainable Course Recommendations for Students at Risk of Dropping out. In A. Mitrovic and N. Bosch, editors, *Proceedings of the 15th International Conference on Educational Data Mining (EDM)*, pages 657–661, Durham, United Kingdom, 2022. International Educational Data Mining Society. https://doi.org/10.5281/zenodo.6853008.

[21] K. Wagner, H. Volkening, S. Basyigit, A. Merceron, P. Sauer, and N. Pinkwart. Which Approach Best Predicts Dropouts in Higher Education?:. In *Proceedings of the 15th International Conference on Computer Supported Education (CSEDU)*, pages 15–26, Prague, Czech Republic, 2023. INSTICC, SciTePress. https://doi.org/10.5220/0011838100003470.

Visual representation of co-authorship with GPT-3: Studying human-machine interaction for effective writing

Antonette Shibani[*]
University of Technology
Sydney, Sydney, Australia
antonette.shibani@uts.edu.au

Ratnavel Rajalakshmi [†]
Vellore Institute of Technology,
Chennai, India
rajalakshmi.r@vit.ac.in

Faerie Mattins
Vellore Institute of Technology,
Chennai, India
faeriemattins.r2019@vitstudent.ac.in

Srivarshan Selvaraj
Vellore Institute of Technology,
Chennai, India
srivarshan.2019@vitstudent.ac.in

Simon Knight
University of Technology
Sydney, Sydney, Australia
simon.knight@uts.edu.au

ABSTRACT

With the recent release of Chat-GPT by OpenAI, the automated text generation capabilities of GPT-3 are seen as transformative and potentially systemically disruptive for higher education. While the impact on teaching and learning practices is still unknown, it is apparent that alongside risks these tools offer the potential to augment human intelligence (intelligence augmentation, or IA). However, strategies for such IA, involving partnership of tool-human, will be needed to support learning. In the context of writing, an investigation of potential approaches is needed given empirical data and studies are currently limited. We introduce a novel visual representation *CoAuthorViz* to examine keystroke logs from a writing assistant where writers interacted with GPT-3 writing suggestions to co-write with the machine. We demonstrate the use of our visualization by exemplifying different kinds of writing behaviour from users writing with GPT-3 support and derive metrics such as their usage of GPT-3 suggestions in relation to overall writing quality indicators. We also release the materials open source to further progress our understanding of desirable user behaviour when working with such state-of-the-art AI tools.

Keywords

Keystroke analysis, Visualization, Writing analytics, GPT-3, Language models, Coauthor, Artificial Intelligence, Chat GPT, Generative AI, CoAuthorViz

1. INTRODUCTION

Changes due to evolving technology is a constant across sectors, but certain technologies have had a profound effect on redefining educational strategies. In academic writing, technologies such as word processing that digitised writing from paper-based formats, the internet and cloud that enabled widespread communication and collaboration, and computational linguistics and Natural language processing that enabled real-time support and automated feedback are key innovations that led to transformations in writing practices and the curriculum [20]. With the recent release of large language models such as Generative Pre-trained Transformer 3 (GPT-3), automated text generation and the use of Artificial Intelligence (AI) to support writing are touted as the next writing transformation.

The open release of powerful tools such as ChatGPT[1] for GPT-3 made visible the dramatic capabilities of generative AI - anyone can write a prompt to ChatGPT in plain English providing instructions, and the tool can generate well-written texts replicating human knowledge. The potential harms and disruptions it can cause to traditional writing curricula have been discussed widely, including concerns about academic integrity, but little is known about how these technologies can best work in practice in partnership with human writers. One such work involves CoAuthor, a human-AI collaborative writing dataset that was created from machine-in-the-loop argumentative and creative writing with writers using automated text suggestions generated from GPT-3 as real-time feedback [21]. The dataset consists of keystroke-level data captured from the writer's typing and is predominantly used by writing analytics and psycholinguistic researchers to learn about cognitive processing. In this paper, we introduce a visual graph CoAuthorViz to aid the analysis of such log data to study human-AI collaboration in writing using more interpretable representations. The intended audience for the CoAuthorViz is researchers who can use the visualisation and related metrics to study the phenomena of working in partnership with AI tools for writing.

2. USING GENERATIVE AI FOR WRITING

Research in the last few decades has seen increasing evidence of the effectiveness of automated writing evaluation (AWE) systems in supporting writers develop their academic writing skills [41] [24] [18]. Automated writing feedback tools

[*] Corresponding author

[†] Corresponding author

[1] https://openai.com/blog/chatgpt/

A. Shibani, R. Rajalakshmi, F. Mattins, S. Selvaraj, and S. Knight. Visual representation of co-authorship with gpt-3: Studying human-machine interaction for effective writing. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 183–193, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115695

provide scalable and innovative computer-based instruction in linguistic, domain, or mixed orientations [14], often targeting specific writing features of interest [18]. However, the most recent advancements in generative AI include the use of large language models for writing, which might fundamentally change how writers learn to write in the future.

Generative Pre-trained Transformer 3 (GPT-3) is a large language model trained on internet data that can automatically generate realistic text [32]. It is a deep learning neural network with over 175 billion machine learning parameters that makes its machine-generated text convincingly similar to what humans write. When a user provides an input text in natural language, the system analyzes the language and predicts the most likely output text. While the beta release of GPT-3 by OpenAI came about much earlier (June 2020), the most recent release of Chat-GPT for public testing in November 2022 has triggered strong reactions to its implications for human writing. Discussions are a mix of initial conversations and scholarly literature given the recency of the topic.

Firstly, we note the potential for GPT-3 usage in writing contexts through applications implemented and evaluated in practice. Automated text generation is the most common application of GPT-3 for generating formal forms of writing, but the model also has the capability to generate poetry, play chess, do arithmetic, translations, and role play, and write code based on user requirements [8]. One use case was seen in 'sparks', sentences generated by the AI writing assistant to inspire writers to create scientific content [13]. The purpose was to aid writers with crafting detailed sentences, providing interesting angles to engage readers, and demonstrating common reader perspectives.

Multiple Intelligent Writing Assistants have made use of GPT-2 and GPT-3 language generation capabilities to help writers develop their content. Examples include writers making integrative leaps in creative writing with multimodal machine intelligence [36], a web application called Wordcraft where users collaborated with a generative model to write a story [42] and a system providing automated summaries to support reflection and revision beyond text generation [9]. A larger evaluation engaging over 60 people to write more than 1,440 stories and essays was performed using CoAuthor, where the interactions between the writer and the GPT-3 suggestions were also captured using keystroke logging [21]. Another writing task that can now be supported by intelligent agents is revision. In the human-in-the loop iterative text revision system called Read, Revise, Repeat (R3), writers interacted with model-generated revisions for deeper edits [11].

However, there are known problems in large language models such as the generation of factually false hallucinations or contradictory information that can exacerbate disinformation [27], bias and immorality arising from human subjectivity [25] and the lack of diversity in its outputs [16]. Perhaps, the more complex problems arising from GPT-3 content relate to social factors such as how it interferes with existing systemic practices affecting people and policies in the real world. There is widespread fear that the automatically generated content amplifies academic dishonesty which is already prevalent in the education sector providing easy means for students to cheat with plagiarism [29]. This is particularly a threat to online learning where the real identity of the writer is hard to discern.

Despite the concerns, there is also hope that these tools might accelerate learning and induce creativity. Like multiple technologies that came before it, some consider these AI tools to be yet another example of humanity's inefficiency dealing with something new that throws their normality into disarray [1]. There is an increasing push to rethink assessments, so we move away from setting assignments that machines can answer towards assessment for learning that captures skills required in the future[33], [39] and students using GPT-3 as part of the curriculum to enhance their learning [30]. There is emerging work such as the launch of 'GPT-2 Output Detector'[2] to identify content authored by Chat-GPT, but with a caveat of having a high false positive rate - dismissing original content as plagiarism could be worse than accepting plagiarised content from the tool for writing assessment. This can be particularly harmful to non-native English writers as GPT detectors may unintentionally penalize writers with constrained linguistic expressions due to their in-built biases [23].

Similar tools and technologies will evolve over time and many students already use AI-based writing tools such as Quillbot[3] as part of their writing practices, so there is an opportunity to investigate how to collaborate with them effectively rather than banning or abolishing them completely [28]. GPT-3 applications where a human stays in the loop are considered safer and the way forward, where the writer uses the machine to augment their writing by utilising its unique capabilities and acknowledges its use [8]. The varied roles AI can take: as an editor, co-author, ghostwriter, and muse have been identified [17], with particular interest towards co-authoring that helps writers develop their writing skills through human-AI partnership [21] that we explore in the current work. Early explorations of two new types of interactions with generative language models show how writers can keep control of their writing by manipulating the auto-generated content [3]. More recent work also involves building a collaborative language model that imitates the entire writing process such as writing drafts, adding suggestions, proposing edits, and providing explanations for its actions, and not just generating the final result [31]. These align with the Intelligence Augmentation (IA) paradigm where human and artificial intelligence work together as a symbiotic system [43], and is of relevance to education where new technology can augment existing teaching and learning strategies [19]. In these cases of co-writing, it is useful to determine the most efficient ways for writers to interact with GPT-3 for optimal partnership and IA, and methods to analyse such behaviour are discussed next.

3. STUDYING WRITING BEHAVIOUR USING KEYSTROKE ANALYSES

Writing is a complex cognitive process that involves recursive and interleaving activities such as planning, translating, reviewing, and monitoring by the writer [12]. Researchers

[2] https://huggingface.co/roberta-base-openai-detector
[3] https://quillbot.com/

use different approaches and data to study the writing process that informs user behaviour. While early work typically relied on resource-intensive manual observation and coding of writing behaviour, computational analysis techniques and log data are now used to study learning processes at scale [10]. These help uncover new patterns from fine-grained information about the learner's writing behaviour through non-obtrusive stealth measurements and keystroke-level data capturing [2][26].

Keystroke logging is a method of automatically capturing data on a user's typing patterns as they write. Analysis of such data can be used to gain insight into various aspects of writing behavior, including typing speed, error rate, and the use of specific keyboard shortcuts. Keystroke analysis has been used for biometric authentication using keystroke dynamics [38], measuring text readability using scroll-based interactions [15], and predicting writing quality for feedback [6]. However, there often exists a disconnect between keystroke level logs and useful insight on cognitive processes that can be derived from it as the data is too fine-grained. Complementary techniques such as eye-tracking and thinking-aloud protocols are often used in combination to capture additional context on the writing [22] [40]. In addition, newer graphic and statistical data analysis techniques offer new perspectives on the writing process.

Visual representations provide a useful starting point to study the complex interactions between sources and writers. Network analysis and graph representations have been used by writing researchers to visualise the temporal development of ideas and links between multiple sources during editing and revising a writer's document [22] [4]. A multi-stage automated revision graph was used to study the evolution of drafts in the revision process that led to the final product and students' interaction with automated feedback based on their frequency of requests [34]. In other work that investigated collaborative writing processes, a revision map was created to represent the joint development of ideas by a group of authors [37]. Such visualisations provide new ways of looking at data to uncover interesting insights and patterns of user behaviour from writing scenarios.

4. OUR WORK

In our work, we introduce a novel visualization called *"CoAuthorViz"* to represent writing behaviours from keystroke logs of users in the CoAuthor dataset (described next). We demonstrate how CoAuthorViz can be used by writing researchers to study co-authorship behaviours of writers interacting with GPT-3 suggestions to co-write with the machine, and investigate metrics derived from such interaction with relation to overall writing quality indicators. We discuss how the work can be extended further to study effective forms of co-authorship with GPT-3 and other AI writing assistants.

4.1 Dataset used

Data for this study comes from the CoAuthor dataset [21] which consists of a total of 1445 writing session data in jsonl format, including 830 creative writing (stories) and 615 argumentative writing (essays) sessions. The dataset contains keystroke-level interactions in a writing session logging 17 events: event name, event source, text delta, cursor range, event timestamp, index of event, a writing prompt to start with, current cursor location, suggestions from GPT-3, number of suggestions to generate per query, the maximum number of tokens to generate per suggestion, sampling temperature, nucleus sampling, presence penalty, and frequency penalty. Descriptions for each variable are provided in the original article [21], and a sample set of rows from the dataset is shown in Table 1. Replays of each individual writing session are also made available on the project website[4].

The writer is provided with an initial prompt by the researchers instructing them to write on the assigned topic, and are required to continue their writing session on their own or with the assistance of GPT-3 sentence recommendations. The writers receive up to five sentence suggestions when a GPT-3 call is made and can do so at any point during their writing sessions - suggestions provided by GPT-3 can be partial or full sentences.

4.2 CoAuthorViz Description

We develop CoAuthorViz to represent co-authorship behaviours of users interacting with GPT-3 suggestions at a sentence-level. This visual representation makes it easier to interpret co-writing processes in comparison to more fine-grained keystroke level logs that capture individual characters and mouse movements. The visualization highlights key actions made by a writer when working with GPT-3 suggestions such as choosing to accept the suggestion as it is, accept suggestion and edit it further, or reject the suggestion and continue writing on their own - these events recorded as part of the keystroke logs can provide significant insight into how AI writing assistants are taken up by writers in practice. Our work is inspired by Automated Revision Graphs previously used for visualising student revision in writing drafts, transferred to the context of co-writing with AI [34].

CoAuthorViz performs sentence-level analysis to visualise interactions between the writer and GPT-3. Three different shapes — circle, triangle, and square represent authorship - the initial prompt provided by researchers is shown as a black circle and ranges from 1 to 9 sentences each (the writer is instructed to base the rest of their writing around it). Since the writer's actual writing starts from the last sentence of the initial prompt, our visualization starts from here. Text entered by the writer is displayed as a gray square, and text written by GPT-3 is displayed as a black triangle. Text modified by the writer after obtaining a GPT-3 suggestion from GPT-3 is displayed as a square overlapping a gray triangle. Empty GPT-3 calls illustrating scenarios where the writer requests for and obtains GPT-3 suggestions, but chooses to ignore them are shown as white triangles. Dotted lines between the shapes indicate a sequence of actions at a sentence level to improve the readability of the visualization and do not have additional meaning.

An example of CoAuthorViz is illustrated in Figure 1. Here, most of the writing was done by the writer independently (see sentences 9, 13, 14, 16-18 with black squares), and even when text from GPT-3 was provided (sentences 8, 10-12, 15 with GPT-3 written text), they went on to add additional text themselves. We also see places where a GPT-3 call was

[4] https://coauthor.stanford.edu/browse/

Table 1: Examples from the dataset with selected rows and columns

eventName	eventSource	textDelta	currentCursor	currentSuggestion
text-insert	user	’ops’: [’retain’: 2017, ’insert’: ’a’]	2017	[]
text-insert	user	’ops’: [’retain’: 2018, ’insert’: ’\n’]	2018	[]
suggestion-get	user	NaN	2019	[]
suggestion-open	api	NaN	2019	[’index’: 0, ’original’: ’smiled at him, and he walked over to her table.’, ’trimmed’: ’Priscilla smiled at him, and he walked over to her table.’, ’probability’: 1.1132658066910296e-05, ’index’: 1, ’original’: ’man walked over to her table and sat down.’, ’trimmed’: ’The man walked over to her table and sat down.’, ’probability’: 1.0074578955483344e-07]
suggestion-hover	user	NaN	2019	[]
suggestion-select	user	NaN	2019	[]
suggestion-close	api	NaN	2019	[]
text-insert	api	’ops’: [’retain’: 2020, ’insert’: ’Priscilla smiled at him, and he walked over to her table.’]	2077	[]

made, but the suggested texts were dismissed and not used by the writer (white triangle in sentences 9, 13, 14, 16-18).

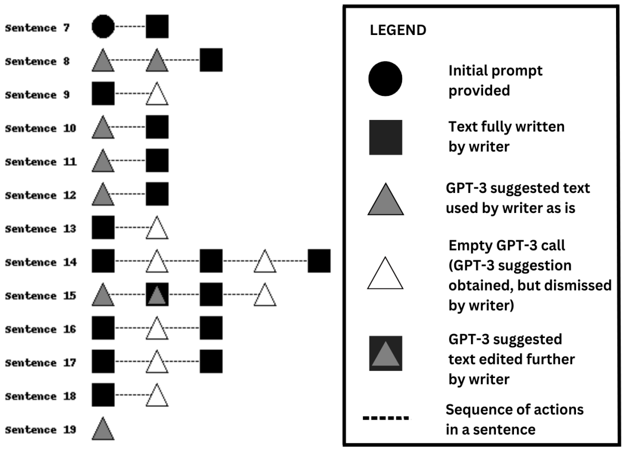

Sentence 7
Sentence 8
Sentence 9
Sentence 10
Sentence 11
Sentence 12
Sentence 13
Sentence 14
Sentence 15
Sentence 16
Sentence 17
Sentence 18
Sentence 19
LEGEND
Initial prompt provided
Text fully written by writer
GPT-3 suggested text used by writer as is
Empty GPT-3 call (GPT-3 suggestion obtained, but dismissed by writer)
GPT-3 suggested text edited further by writer
Sequence of actions in a sentence

Figure 1: Example of a CoAuthorViz with descriptors

CoAuthorViz generates a simple visualization to represent co-authorship with GPT-3 from relatively complex, fine-grained keystroke-level data. It reveals insights on the writer’s frequency of autonomous writing without AI assistance and their usage, dismissal, and modification of GPT-3 text suggestions provided. These can be used to inform the study of user behaviour when engaging with AI writing assistants such as GPT-3.

4.3 Technical Implementation

The lack of standards in capturing and analysing keystroke data is an identified challenge in this kind of research [22]. To this end, we provide a detailed explanation of the construction of CoAuthorViz and release the materials open source (including the scripts and plots generated) to help facilitate knowledge exchange among research groups [Github link].

The keystroke log is first read by iterating over all the tracked events. Text at any given keystroke is rebuilt from the log using events and cursor positions. This is done by maintaining a text buffer during the entire process providing the current state of the document - when a text insertion keystroke is encountered, the corresponding text is added to the buffer; when text deletion occurs, the corresponding characters are deleted from the buffer; cursor positions are used to identify the locations in the buffer when such events occur. The events and their corresponding text buffers are grouped by the number of sentences in the buffer, providing a sequence of all events at the sentence level. From this sentence-level event sequence, the following steps are performed to define key constructs of interest:

1. **GPT-3 Suggestion Selection:** “suggestion-get” events that are succeeded by a “suggestion-select” event are identified as GPT-3 calls where the writer obtained a suggestion and made use of it. Related “suggestion-open”, “suggestion-hover”, “suggestion-down”, “suggestion-up”, and “suggestion-reopen” events are removed as they are all indicative of the same event - author choosing from the GPT-3 suggestions. “text-insert” events occurring immediately after the “suggestion-select” events are removed as they also signify the insertion of GPT-3 suggestion selected by the writer

2. **Empty GPT-3 Call:** “suggestion-get” events that do not have a succeeding selection event are identified as empty GPT-3 calls where the author did not incorporate any suggestion provided by GPT-3

3. **GPT-3 Suggestion Modification:** Any “cursor-backward”, “cursor-select” or “text-delete” events immediately suc-

ceeding a "suggestion-select" event, but without any "text-insert" event in between are perceived as modifications done by the author to the GPT-3 suggestion they chose. All cursor movement events, text deletion events and "suggestion-close" events are removed

4. **User Text Addition:** Consecutive "text-insert" events are grouped for piecing together text written by the writer

Metrics are calculated by counting the key events in relation to GPT-3 calls, and authorship in sentences. The sequence of key identified events from the above constructs is generated as a visualisation using the Pillow package [5]. The full implementation runs on a Python notebook, and is represented in Figure 2.

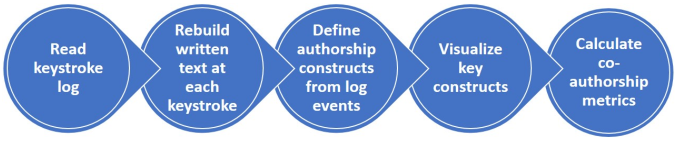

Read keystroke log
Rebuild written text at each keystroke
Define authorship constructs from log events
Visualize key constructs
Calculate co-authorship metrics

Figure 2: Steps in the construction of CoAuthorViz

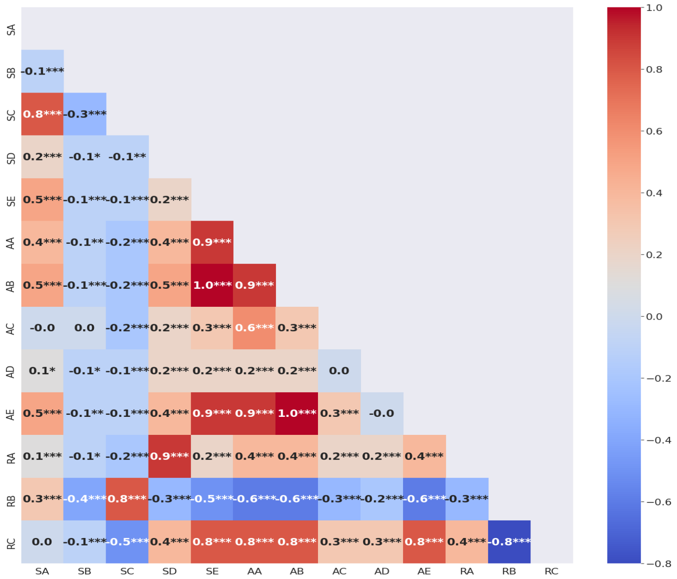


Figure 3: Correlation matrix with statistical significance of CoAuthorViz metrics

5. FINDINGS AND DISCUSSION

In this section, we discuss the main findings from our visualisation and examine sample cases in detail demonstrating the application of CoAuthorViz for researching writing.

5.1 Analysis of CoAuthorViz metrics

A summary of the key events noted in CoAuthorViz is generated for each writing session providing tangible metrics that can be studied along with the visualization. Three types: Sentence level, API-based, and Ratio metrics are provided - see Table 2 for the summary statistics of these metrics. Each of the 1445 writing sessions in the CoAuthor generates a total number of sentences ranging from 11 to 78, and an average of 29 sentences in the final writing. The initial prompt in a writer's writing session can vary from 0 to 9 sentences, with an average of around 4. The number of sentences in the initial prompt is 0 in cases where the writer deletes the initial prompt and rewrites it from scratch.

Metrics on the number of sentences written entirely by the writer, GPT-3, or a combination of the writer and GPT-3 are populated. Additional metrics include the frequency of using GPT-3 suggestions with and without modification, as well as the number of instances where a GPT-3 call was made but the suggestion was rejected, likely because the writer was dissatisfied with the suggested texts. Ratios were also calculated to characterize GPT-3 versus writer authoring in relation to the total number of sentences generated in a writing session.

From the summary statistics table in Table 2, we derive insights on the usage of GPT-3 across the 1445 writing sessions. The average number of times GPT-3 calls were made (AA) was 12.5 but varied widely across the sessions (SD = 9.2) with a minimum of 0 and a maximum of 65. Similarly, there was high variance in the number of times a GPT-3 suggestion was incorporated (AB) ranging from 0 to 47 (M = 8.9, SD = 7.4), and the number of times a GPT-3 suggestion was accepted as it is (AE) (M = 7.3, SD = 7.2). Total GPT-3 usage in their sentences (RC) was calculated from the ratio of the sum of sentences using GPT-3 suggestion, and the total number of sentences in the writing ranged from 0 to 0.87 (M = 0.3, SD = 0.2). The ratio of the number of times the suggestion is rejected to the number of times the author calls for GPT-3 (AC/AA) indicates that suggestions made by GPT-3 were rejected 29.31% of the time, and suggestions were accepted as is 58% of the time (AE/ AA).

We also calculate correlation to examine relations within CoAuthorViz metrics. Figure 3 shows the matrix of Pearson correlation coefficients (CC) for each pair of metrics in the summary table. The statistical significance of each correlation is indicated by the number of asterisks adjacent to the value (in order of increasing significance: p-value < 0.05 is flagged with one star (*), p-value < 0.01 is flagged with 2 stars (**), and p-value < 0.001 is flagged with three stars (***)). Related pairs of metrics such as AA and AE have high CC ranging from 0.8 to 1.0 because the metrics are computed from similar underlying values such as the number of GPT-3 calls made.

A negative correlation (r = -0.6) was found between the autonomous writing indicator (RB) and the number of times a GPT-3 suggestion is accepted as is (AE). Similarly, writers having high GPT-3 dependence indicators had more sentences completely authored by GPT-3 (r = 0.9) suggesting their reliance on GPT-3 for writing without making further edits. On the contrary, writers who had a high number of sentences completely authored by them preferred to write their sentences independent of GPT-3 and hence tended to have high autonomous writing indicators (r = 0.8). The total number of GPT-3 calls made positively correlated to both the number of times its suggestion was accepted as is (r =0.9) and the number of sentences co-authored by GPT-3 and the writer (r = 0.9).

Table 2: Summary Statistics of CoAuthorViz metrics

Type	Metrics (for sample size n=1445)	Mean	Median	Standard Deviation	Min	Max
Sentence Metrics	Total number of sentences **(SA)**	28.962	27	10.388	11	78
	Number of sentences in initial prompt **(SB)**	4.421	4	2.390	0	9
	Number of sentences completely authored by the writer **(SC)**	16.242	15	9.535	0	64
	Number of sentences completely authored by GPT-3 **(SD)**	0.685	0	1.886	0	22
	Number of sentences co-authored by GPT-3 and writer **(SE)**	7.613	6	5.953	0	42
API Metrics	Total number of GPT-3 calls made **(AA)**	12.531	10	9.204	0	65
	Number of times GPT-3 suggestion is accepted **(AB)**	8.857	7	7.424	0	47
	Number of times writer rejected GPT-3 suggestion **(AC)**	3.673	3	3.530	0	24
	Number of times GPT-3 suggestion is modified **(AD)**	1.586	1	1.796	0	10
	Number of times GPT-3 suggestion is accepted as it is **(AE)**	7.271	5	7.233	0	47
Ratio Metrics	GPT-3 dependence indicator - Number of sentences completely authored by GPT-3 / Total number of sentences **(RA)**	0.021	0	0.054	0	0.611
	Autonomous writing indicator - Number of sentences completely authored by writer / Total number of sentences **(RB)**	0.541	0.564	0.205	0	0.962
	Total GPT-3 usage in sentences [(SD+SE)/SA] **(RC)**	0.285	0.25	0.183	0	0.867
TAACO Metrics	lemma_ttr **(LTTR)**	0.401	0.4	0.054	0.240	0.585
	adjacent_overlap_all_sent **(AOAS)**	0.212	0.210	0.043	0.076	0.389
	adjacent_overlap_all_para **(AOAP)**	0.256	0.258	0.090	0.0	0.863
	lsa_1_all_sent **(LSA1AS)**	0.309	0.305	0.094	0.094	0.688
	lsa_1_all_para **(LSA1AP)**	0.477	0.490	0.171	0.0	0.948
	all_connective **(AP)**	0.068	0.067	0.017	0.017	0.131

Table 3: t-test results for TAACO Metrics with alpha value as 0.025 and degree of freedom as 1444.

Metrics	Low GPT-3 usage Group		High GPT-3 usage Group		T-Statistic	P-Value
	Mean	Standard Deviation	Mean	Standard Deviation		
LTTR	0.406	0.052	0.396	0.056	3.592	3.386×10^{-4}
AOAS	0.203	0.040	0.220	0.044	-7.787	1.298×10^{-14}
AOAP	0.259	0.084	0.253	0.096	1.360	1.739×10^{-1}
LSA1AS	0.307	0.094	0.312	0.093	-1.099	2.716×10^{-1}
LSA1AP	0.488	0.161	0.466	0.180	2.432	1.511×10^{-2}
AP	0.068	0.016	0.068	0.017	0.472	6.363×10^{-1}

5.2 Relation between CoAuthorViz metrics and writing features

We additionally analysed the final written texts from the CoAuthor sessions using TAACO to derive indicators of writing quality from language features [7]. Key indicators of lexical diversity, lexical overlap, semantic overlap, and connectedness below were used to derive the metrics, and include descriptions from TAACO on how the metrics are calculated:

- Lemma_ttr (LTTR) - number of unique lemmas (types) divided by the number of total running lemmas (tokens)
- Adjacent_overlap_all_sent (AOAS) - number of lemma types that occur at least once in the next sentence
- Adjacent_overlap_all_para (AOAP) - number of lemma types that occur at least once in the next paragraph
- Lsa_1_all_sent (LSA1AS) - Average latent semantic analysis cosine similarity between all adjacent sentences
- Lsa_1_all_para (LSA1AP) - Average latent semantic analysis cosine similarity between all adjacent paragraphs
- All_connective (AP) - number of all connectives

The above TAACO metrics were used for preliminary analysis of our visualization metrics in relation to writing quality features since the CoAuthor dataset did not contain a quality metric for the text outputs from the writing sessions - the correlation matrix is shown in Figure 4. However, we do not see a significant correlation between any CoAuthorViz metric and automated writing features extracted from TAACO.

We further split session users into two groups based on the number of GPT-3 calls initiated to study potential differences between groups. Sessions with the total number of GPT-3 calls above or equal to the median value were classified as belonging to the high GPT-3 usage group and below median sessions formed the low GPT-3 usage group. We performed a t-test (Findings in Table 3) to compare TAACO metrics between the high GPT-3 usage group (n = 718) and the low GPT-3 usage group (n = 728).

Results suggest that there was a significant difference in

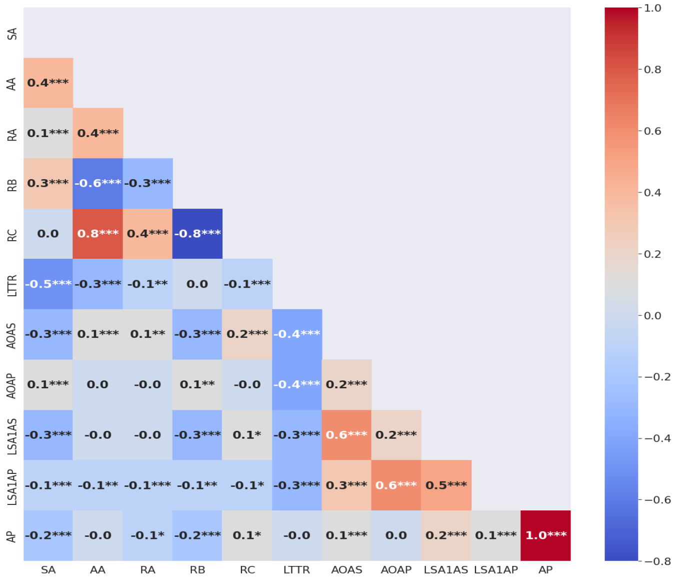


Figure 4: Correlation matrix with statistical significance of CoAuthorViz and TAACO metrics

Lemma type-token ratio (LTTR) between the high usage group (M = 0.396, SD = 0.057) and the low usage group (M = 0.406, SD = 0.053); t(df=1444) = 3.6, p < .005, meaning that writers who accessed GPT-3 less produced a higher proportion of the text that consisted of content words (nouns, lexical verbs, adjectives, and adverbs derived from adjectives) indicating higher lexical diversity. An opposite effect was observed for the TAACO metric Adjacent sentence overlap all lemmas (AOAS) between the high usage group (M = 0.221, SD = 0.045) and the low usage group (M = 0.203, SD = 0.041); t(df=1444) = -7.8, p < .005, suggesting that writings from the high GPT-3 usage group had higher lexical overlaps in adjacent sentences leading to more cohesion.

A significant difference was also observed in Lsa cosine similarity in adjacent paragraphs (LSA1AP) between the high usage group (M = 0.467, SD = 0.18) and the low usage group (M = 0.489, SD = 0.162); t(df=1444) = 2.4, p = .02. Here, writing from the low GPT-3 usage group had a higher semantic overlap exhibiting high average latent semantic analysis cosine similarity between all adjacent paragraphs. A descriptive box plot showing the minimum, maximum, median, lower, and upper quartiles of the three metrics in the high and low groups is shown in Figure 5. No significant difference in group means was noted for the other three metrics (AOAP, LSA1AS, and AP). While the findings indicate effects of high/ low GPT-3 usage in the output writing produced, higher level features are required in order to draw stronger links to writing quality, likely using some form of human assessment in the future.

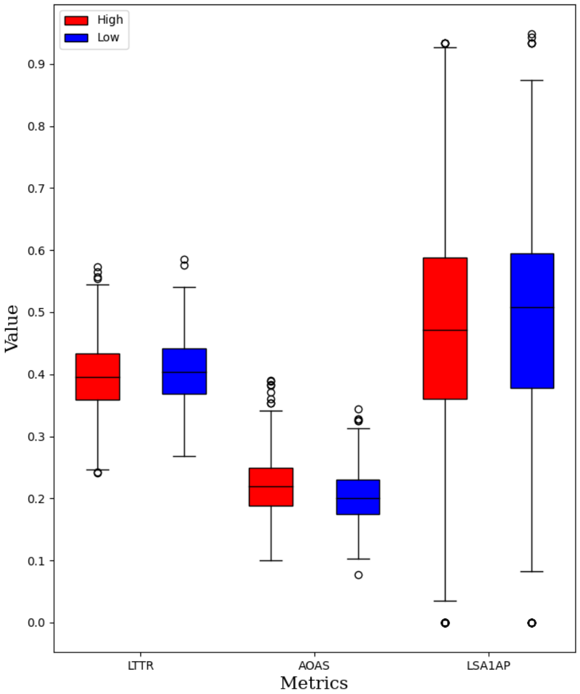


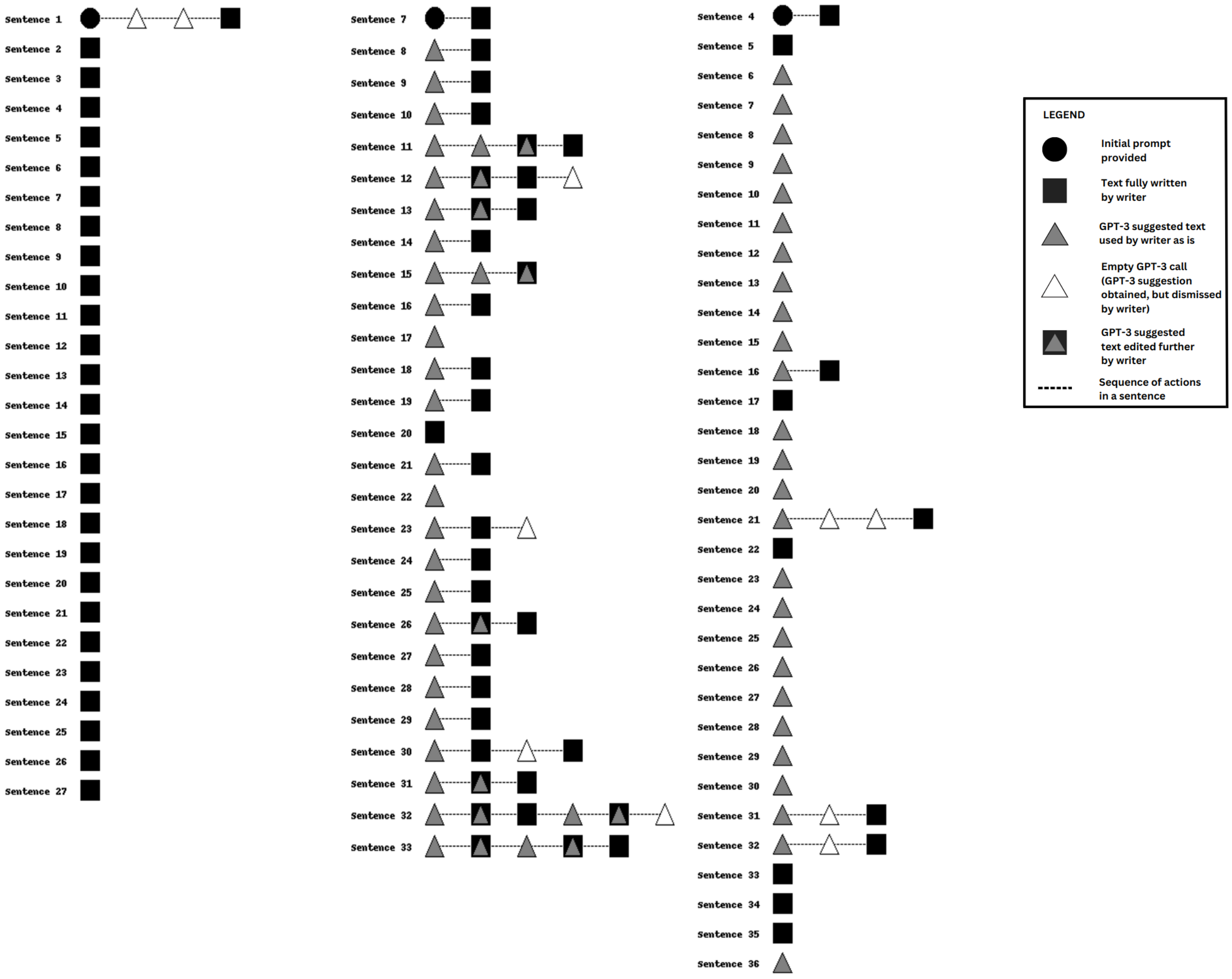
Figure 5: Box plots describing differences in TAACO Metrics for the high and low GPT-3 usage groups

5.3 Case studies of writer interaction with GPT-3 for co-authorship

We further demonstrate the use of CoAuthorViz to study in detail writer interactions with GPT-3 using example writing sessions. We show three cases from the dataset in Figure 6 showcasing differences in writers' behaviour when working with GPT-3 suggestions on their writing. Metrics from these writing sessions are shown in Table 4.

Table 4: Summary table for the writing session shown in 6.

Metrics	Case-1	Case-2	Case-3
SA	27	33	36
SB	1	7	4
SC	26	1	6
SD	0	2	22
SE	0	23	4
AA	2	33	30
AB	0	29	26
AC	2	4	4
AD	0	10	0
AE	0	19	26
RA	0.0	0.060	0.611
RB	0.962	0.030	0.166
RC	0.0	0.757	0.722
LTTR	0.383	0.389	0.308
AOAS	0.290	0.186	0.295
AOAP	0.354	0.218	0.0
LSA1AS	0.392	0.409	0.423
LSA1AP	0.532	0.535	0.0
AP	0.077	0.083	0.105



(a) Case-1: Fully autonomous writer

(b) Case-2: Autonomous writer with GPT-3 assistance

(c) Case-3: GPT-3 dependent writer

Figure 6: Sample cases of user's writing sessions demonstrated using CoAuthorViz

5.3.1 *Case 1: Fully autonomous writer*

The first sample session illustrated in Figure 6a illustrates an example where the writer is completely autonomous and decides not to use any GPT-3 suggestions in their writing. Starting to write from the initial prompt in sentence 1, the writer makes two GPT-3 calls but rejects its suggestions and decides to write by themselves thereon. The writer was perhaps not satisfied with the sentence suggestions offered by GPT-3 and decided not to get any more suggestions from it to not waste their time further. Table 4 shows that this session's autonomous writing indicator (RB = 0.96) is very high.

5.3.2 *Case 2: Autonomous writer with GPT-3 assistance*

The second case shown in Figure 6b shows an example where the writer incorporates a lot of GPT-3 suggestions in their writing, but modifies the sentences to suit their writing style. They start to write following the 7-sentence prompt provided and frequently get suggestions from GPT-3. In ten instances, the writer modifies the GPT-3 suggestion provided (overlapping triangle and square in sentences 11-13, 15, 26, 31-33) and in over 15 instances, they go on to add their own phrasing in addition to GPT-3 sentence suggestions (Sentences 8-10, 14, 16, 18, 19, 21, 23-30). Even though the autonomous writing indicator is low (RB = 0.03) for this session (because it is influenced by the number of sentences completely authored by the writer), we observe that throughout the entire writing session, while they get assistance from GPT-3, the writer still demonstrates some autonomy in their writing by adding text on their own or modifying the GPT-3 suggestion. This is a great example of the potentially optimal use of machine assistance in combination with the writer's own writing and intelligence augmentation [43]. From Table 4, we observe that LTTR is 0.389, which is the highest of all three cases - the writing generated with GPT-3 assistance exhibited more diverse vocabulary [21].

5.3.3 Case 3: GPT-3 dependent writer

The final case illustrated in Figure 6c depicts the case of a writer who primarily used GPT-3 to create their piece of writing. Here, the writer starts off by adding sentences of their own 4 and 5 (following the initial prompt containing 4 sentences), before they become heavily dependent on GPT-3 for suggestions. The GPT-3 dependence indicator (RA) was 0.611 and the autonomous writing indicator (RB) was 0.166, evidencing that a considerable part of their writing was written by GPT-3. However, note that the writer demonstrated some autonomy by modifying GPT-3 suggestions, likely because they did not find them suitable (Sentences 18, 21, 31, and 32) and authored a few sentences themselves (Sentences 17, 21, 22, 31-35). This example demonstrates a writing style where the writer relied on GPT-3 suggestions repeatedly and used the system to its full advantage. The LTTR, in this case, is the lowest of the three cases (0.308) - there is less diverse vocabulary in this writing in comparison to both the autonomous writing by the writer in case 1 and GPT-assisted writing in case 2.

6. CONCLUSION

The paper introduced a novel approach to studying the coauthorship behaviour of writers interacting with GPT-3, a recent artificial intelligence (AI) tool producing auto-generated content. Keystroke logs from users' writing sessions in *CoAuthor* [21], where writers used automated text suggestions generated from GPT-3 as real-time feedback formed the basis of our analysis. Empirical studies on user interaction with GPT-3 are limited - this research fills the gap by introducing new methods of analysis and demonstrating diverse user behaviour when interacting with generative AI. The insights are also derived at an interpretable level for researchers building on keystroke data containing low-level details such as the character entered, current cursor location, etc. which is hard to read.

We developed 'CoAuthorViz', a visualization to represent interactions between the writer and GPT-3 at a sentence level - this captured key constructs such as the writer incorporating a GPT-3 suggested text as is (GPT-3 suggestion selection), the writer not incorporating a GPT-3 suggestion (Empty GPT-3 call), the writer modifying the suggested text (GPT-3 suggestion modification), and the writer's own writing (user text addition). Three different sample cases of writing exhibiting full autonomy in writing, using GPT-3 for assistance and GPT-3 dependence were shown to demonstrate the use of CoAuthorViz to study writing behaviours.

We derived additional CoAuthorViz metrics such as a GPT-3 dependence indicator, an autonomous writing indicator, and other GPT-3 suggestion incorporation metrics to quantify human and AI authorship. The average number of GPT-3 calls across the 1445 writing sessions was 12.5, but varied widely across the sessions (SD = 9.2). Automated sentence suggestions from GPT-3 were accepted as is 58% of the time and suggestions were rejected 29.31 % of the time, indicative of diverse writing behaviours with respect to interaction with GPT-3. Statistical analysis on CoAuthorViz metrics in relation to overall writing quality indicators derived from TAACO [7] showed that writers who accessed GPT-3 less produced writing with higher lexical density (more content words) and higher semantic overlap (higher average latent semantic analysis cosine similarity between all adjacent paragraphs). While the results showed the effects of high/ low GPT-3 usage in the output writing in terms of selected linguistic features, higher-level features are required to draw stronger links to writing quality. This can be done in the future by manually assessing the writing produced by the two groups of writers using a standard rubric for writing assessment.

From the three sample cases illustrated, we observed varied levels of autonomy exhibited by the writer when incorporating GPT-3 suggestions in their writing. These insights are useful for writing researchers to understand cognitive writing processes involved in human-AI partnerships from rich and nuanced log data. This could be the first step towards developing visual analytics that might be intelligible to a trained instructor grading the writing, or the basis for automated textual feedback to the instructor and/or student to improve their writing practices. We aim to further examine CoAuthorViz and its metrics for investigating comparable traits across different groups of writers and provide feedback for effective engagement. By studying effective user behaviours for enhanced human-AI partnership in writing, we can better understand how intelligence augmentation can be achieved in practice through critical engagement [43] [35].

The general consensus is that a partnership between the machine and the human is desirable for learning [28], but we need to understand and define what an *optimal partnership* is when working with generative AI for intelligence augmentation. There still remain questions on what constitutes desirable behaviours when it comes to interaction with GPT-3 - Is more autonomy (in terms of self-writing and edits to GPT-3) considered more optimal? Is it the one producing a better piece of writing irrespective of the repetitive use of GPT-3 and dependence? Do writers require foundational knowledge and skills to use AI tools to critique and use them appropriately? Do AI tools supplant critical processes and thinking that the learner ought to develop? These questions need further investigation.

Issues related to academic integrity also need due consideration. How one should attribute GPT-3 usage when coauthoring pieces of writing, and to what level is GPT-3 usage acceptable are open questions. In addition, the question of fairness remains as students who get access to better AI tools might be able to produce better writing [28] - accessibility issues may be elevated when these tools start to be distributed by companies for commercial profit at the end of public evaluation periods. With continuing advances in the intersection of technology, research, and practice, AI-augmented writing should enrich human knowledge for all.

7. REFERENCES

[1] B. Alexander. Chatgpt and higher education: last week and this week. https://bryanalexander.org/future-trends-forum/chatgpt-and-higher-education-last-week-and-this-week/, 2022. Accessed: 2023-01-11.

[2] L. K. Allen, M. E. Jacovina, M. Dascalu, R. D. Roscoe, K. M. Kent, A. D. Likens, and D. S. McNamara. {ENTER} ing the time series {SPACE}: Uncovering the writing process through keystroke analyses. *International Educational Data Mining Society*, 2016.

[3] K. C. Arnold, A. M. Volzer, and N. G. Madrid. Generative models can help writers without writing for them. In , editor, *IUI Workshops*, volume 2903, pages 1–8, United States, 2021. CEUR Workshop Proceedings.

[4] G. Caporossi and C. Leblay. Online writing data representation: A graph theory approach. In J. Gama, E. Bradley, and J. Hollmén, editors, *Advances in Intelligent Data Analysis X*, pages 80–89, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[5] A. Clark et al. Pillow (pil fork) documentation. *readthedocs*, 2015.

[6] R. Conijn, C. Cook, M. van Zaanen, and L. Van Waes. Early prediction of writing quality using keystroke logging. *International Journal of Artificial Intelligence in Education*, 32(4):835–866, 2022.

[7] S. A. Crossley, K. Kyle, and D. S. McNamara. The tool for the automatic analysis of text cohesion (taaco): Automatic assessment of local, global, and text cohesion. *Behavior research methods*, 48(4):1227–1237, 2016.

[8] R. Dale. Gpt-3: What's it good for? *Natural Language Engineering*, 27(1):113–118, 2021.

[9] H. Dang, K. Benharrak, F. Lehmann, and D. Buschek. Beyond text generation: Supporting writers with continuous automatic text summaries. UIST '22, pages 1–13, New York, NY, USA, 2022. Association for Computing Machinery.

[10] P. Deane, N. Odendahl, T. Quinlan, M. Fowles, C. Welsh, and J. Bivens-Tatum. Cognitive models of writing: Writing proficiency as a complex integrated skill. *ETS Research Report Series*, 2008(2):1 – 36, 2008.

[11] W. Du, Z. M. Kim, V. Raheja, D. Kumar, and D. Kang. Read, revise, repeat: A system demonstration for human-in-the-loop iterative text revision. *arXiv preprint arXiv:2204.03685*, 2022.

[12] L. Flower and J. R. Hayes. The cognition of discovery: Defining a rhetorical problem. *College composition and communication*, 31(1):21–32, 1980.

[13] K. I. Gero, V. Liu, and L. Chilton. Sparks: Inspiration for science writing using language models. In *Designing Interactive Systems Conference*, pages 1002–1019, 2022.

[14] A. Gibson and A. Shibani. Natural language processing-writing analytics. *by Charles Lang, George Siemens, Alyssa Friend Wise, Dragan Gašević, and Agathe Merceron. 2nd ed. Vancouver, Canada: SoLAR*, pages 96–104, 2022.

[15] S. Gooding, Y. Berzak, T. Mak, and M. Sharifi. Predicting text readability from scrolling interactions. *arXiv preprint arXiv:2105.06354*, 2021.

[16] D. Ippolito, R. Kriz, M. Kustikova, J. Sedoc, and C. Callison-Burch. Comparison of diverse decoding methods from conditional language models. *arXiv preprint arXiv:1906.06362*, 2019.

[17] G. M. Kleiman and GPT-3. Ai in writing class: Editor, co-author, ghostwriter, or muse? https://medium.com/@glenn_kleiman/ai-in-writing-class-editor-co-author-ghostwriter-or-muse-348532d896a6, 2022. Accessed: 2023-01-20.

[18] S. Knight, A. Shibani, S. Abel, A. Gibson, P. Ryan, N. Sutton, R. Wight, C. Lucas, A. Sandor, K. Kitto, et al. Acawriter: A learning analytics tool for formative feedback on academic writing. 2020.

[19] S. Knight, A. Shibani, and S. Buckingham-Shum. Augmenting formative writing assessment with learning analytics: A design abstraction approach. International Society of the Learning Sciences, Inc.[ISLS]., 2018.

[20] O. Kruse, C. Rapp, C. Anson, K. Benetos, E. Cotos, A. Devitt, and A. Shibani. Analytics techniques for analysing writing. In , editor, *Digital Writing Technologies in Higher Education: Theory, Research, and Practice*, pages 0–0. Springer, Berlin, Germany, 2023. In Submission.

[21] M. Lee, P. Liang, and Q. Yang. CoAuthor: Designing a human-AI collaborative writing dataset for exploring language model capabilities. In *CHI Conference on Human Factors in Computing Systems*, pages 1–19. ACM, apr 2022.

[22] M. Leijten and L. Van Waes. Keystroke logging in writing research: Using inputlog to analyze and visualize writing processes. *Written Communication*, 30(3):358–392, 2013.

[23] W. Liang, M. Yuksekgonul, Y. Mao, E. Wu, and J. Zou. Gpt detectors are biased against non-native english writers, 2023.

[24] S. Link, M. Mehrzad, and M. Rahimi. Impact of automated writing evaluation on teacher feedback, student revision, and writing improvement. *Computer Assisted Language Learning*, 35(4):605–634, 2022.

[25] L. Lucy and D. Bamman. Gender and representation bias in gpt-3 generated stories. In *Proceedings of the Third Workshop on Narrative Understanding*, pages 48–55, 2021.

[26] D. Malekian, J. Bailey, G. Kennedy, P. de Barba, and S. Nawaz. Characterising students' writing processes using temporal keystroke analysis. *International Educational Data Mining Society*, 2019.

[27] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*, 2020.

[28] L. McKnight. In an ai world we need to teach students how to work with robot writers. https://theconversation.com/in-an-ai-world-we-need-to-teach-students-how-to-work-with-robot-writers-157508/, 2021. Accessed: 2023-01-11.

[29] S. E. E. Michael Mindzak. Artificial intelligence is getting better at writing, and universities should worry about plagiarism. https:

//theconversation.com/artificial-intelligence-is-getting-better-at-writing-and-universities-should-worry-about-plagiarism-160481, 2021. Accessed: 2023-01-11.

[30] E. R. Mollick and L. Mollick. New modes of learning enabled by ai chatbots: Three methods and assignments. *SSRN Electronic Journal*, 0:1–21, 2022.

[31] T. Schick, J. A. Yu, Z. Jiang, F. Petroni, P. Lewis, G. Izacard, Q. You, C. Nalmpantis, E. Grave, and S. Riedel. PEER: A collaborative language model. In *The Eleventh International Conference on Learning Representations*, pages 1 – 24, 2023.

[32] R. Schmelzer. Gpt-3. https://www.techtarget.com/searchenterpriseai/definition/GPT-3, 2021. Accessed: 2023-01-11.

[33] M. Sharples. New ai tools that can write student essays require educators to rethink teaching and assessment. *Impact of Social Sciences Blog*, 2022.

[34] A. Shibani. Constructing automated revision graphs: A novel visualization technique to study student writing. In *International Conference on Artificial Intelligence in Education*, pages 285–290. Springer, 2020.

[35] A. Shibani, S. Knight, and S. Buckingham Shum. Questioning learning analytics? cultivating critical engagement as student automated feedback literacy. In *LAK22: 12th International Learning Analytics and Knowledge Conference*, pages 326–335, 2022.

[36] N. Singh, G. Bernal, D. Savchenko, and E. L. Glassman. Where to hide a stolen elephant: Leaps in creative writing with multimodal machine intelligence. *ACM Transactions on Computer-Human Interaction*, 0:0–0, 2022. Just Accepted.

[37] V. Southavilay, K. Yacef, P. Reimann, and R. A. Calvo. Analysis of collaborative writing processes using revision maps and probabilistic topic models. In *Proceedings of the third international conference on learning analytics and knowledge*, pages 38–47, 2013.

[38] Y. Sun, H. Ceker, and S. Upadhyaya. Shared keystroke dataset for continuous authentication. In , editor, *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, Abu Dhabi, United Arab Emirates, 2016. IEEE.

[39] Z. Swiecki, H. Khosravi, G. Chen, R. Martinez-Maldonado, J. M. Lodge, S. Milligan, N. Selwyn, and D. Gašević. Assessment in the age of artificial intelligence. *Computers and Education: Artificial Intelligence*, 3:100075–100085, 2022.

[40] Å. Wengelin, M. Torrance, K. Holmqvist, S. Simpson, D. Galbraith, V. Johansson, and R. Johansson. Combined eyetracking and keystroke-logging methods for studying cognitive processes in text production. *Behavior research methods*, 41(2):337–351, 2009.

[41] J. Wilson and R. D. Roscoe. Automated writing evaluation and feedback: Multiple metrics of efficacy. *Journal of Educational Computing Research*, 58(1):87–125, 2020.

[42] A. Yuan, A. Coenen, E. Reif, and D. Ippolito. Wordcraft: Story writing with large language models. In *27th International Conference on Intelligent User Interfaces*, IUI '22, page 841–852, New York, NY, USA, 2022. Association for Computing Machinery.

[43] L. Zhou, S. Paul, H. Demirkan, L. Yuan, J. Spohrer, M. Zhou, and J. Basu. Intelligence augmentation: towards building human-machine symbiotic relationship. *AIS Transactions on Human-Computer Interaction*, 13(2):243–264, 2021.

To speak or not to speak, and what to speak, when doing task actions collaboratively*

Jauwairia Nasir†
University of Augsburg
jauwairia.nasir@uni-a.de

Aditi Kothiyal†
Indian Institute of Technology Gandhinagar
aditi.kothiyal@iitgn.ac.in

Haoyu Sheng
École polytechnique fédérale de Lausanne
haoyu.sheng@epfl.ch

Pierre Dillenbourg
École polytechnique fédérale de Lausanne
pierre.dillenbourg@epfl.ch

ABSTRACT

Transactive discussion during collaborative learning is crucial for building on each other's reasoning and developing problem solving strategies. In a tabletop collaborative learning activity, student actions on the interface can drive their thinking and be used to ground discussions, thus affecting their problem-solving performance and learning. However, it is not clear how the interplay of actions and discussions, for instance, how students performing actions or pausing actions while discussing, is related to their learning. In this paper, we seek to understand how the transactivity of actions and discussions is associated with learning. Specifically, we ask what is the relationship between discussion and actions, and how it is different between those who learn (gainers) and those who do not (non-gainers). We present a combined differential sequence mining and content analysis approach to examine this relationship, which we applied on the data from 32 teams collaborating on a problem designed to help them learn concepts of minimum spanning trees. We found that discussion and action occur concurrently more frequently among gainers than non-gainers. Further we find that gainers tend to do more reflective actions along with discussion, such as looking at their previous solutions, than non-gainers. Finally, gainers discussion consists more of goal clarification, reflection on past solutions and agreement on future actions than non-gainers, who do not share their ideas and cannot agree on next steps. Thus this approach helps us identify how the interplay of actions and discussion could lead to learning, and the findings offer guidelines to teachers and instructional designers regarding indicators of productive collaborative learning, and when and how, they should intervene to improve learning. Concretely, the results suggest that teachers should support elaborative, reflective and planning discussions along with reflective actions.

Keywords

transactivity, sequence mining, content analysis, collaborative learning

1. INTRODUCTION

When students collaborate to learn from computer supported collaborative learning (CSCL) environments, their learning depends not only on the quality of their interaction with each other, but also with the learning activity [5]. In other words, students need to align both in terms of their activities (such as problem-solving steps or co-writing) and their discussion (of strategies and knowledge) [23]. Specifically, in CSCL environments that involve problem-solving to build conceptual understanding, for the activity to be effective for learning, team members need to develop a joint problem space and construct knowledge through the process of explanation, negotiation and mutual regulation [24, 5]. To achieve this, their actions must either move them towards the solution, or provide them some information that generates potential and motivates future problem-solving actions [25]. Actions can thus help ground collaboration [2], if they are followed with the right kind of discussion, i.e., students discussions should then build on and leverage this information or potential to further understand the problem, decide on next steps and construct meaning from the problem-solving experience [24, 6]. Thus, discussion and actions together play critical roles in problem-solving CSCL environments, as it is through both these means that students obtain and share task-related information to build a common ground, develop problem-solving or learning strategies and regulate their learning. For instance, as described in [4] children's body movements and task-related speech evolve together and serve the purposes of communication and co-ordination, and as cognitive tools for knowledge construction. Similarly, research has also shown that acting together on task-related objects accompanied with speech was related to effective collaboration [15].

[†] Both authors contributed equally to this work and share first authorship. A majority of the work was conducted while the authors were employed at CHILI Lab, EPFL

[*] (Does NOT produce the permission block, copyright information nor page numbering). For use with edm_article.cls.

J. Nasir, A. Kothiyal, H. Sheng, and P. Dillenbourg. To speak or not to speak, and what to speak, when doing task actions collaboratively. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 194–206, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115689

Within the collaborative learning research space, transactivity or student's discussion that builds on each other's reasoning by interpreting team member's statements, asking questions, extending, critiquing and integrating has been used as a metric to evaluate the effectiveness of collaboration [29, 28]. We argue that in synchronous problem-solving CSCL environments, the notion of transactivity should be extended to discussion and actions together, i.e., actions that build on students' discussion, and discussions that build on actions. For instance, students should explicate the information gained from an action such that team members can then discuss about what this suggests for the next problem-solving steps [6]. However, not all actions need be accompanied with discussions. For instance, students may plan and perform a set of actions, or they may perform and reflect on each action [18]. The key question then is, should teams discuss while performing actions, i.e., building on each other's ideas 'on the go' or should they 'stop and pause' their actions to discuss their ideas, or both? Further, how are each of these behaviours related to learning? Finally, which kind of discussion accompanying actions is productive? The answers of these questions are necessary to support teachers in intervening at the right time to guide students actions and discussions, or in the design of feedback built into CSCL environments.

Previous research that analysed students' discussion and actions together in an attempt to identify joint discussion and action indicators of collaboration followed one of two approaches. The first one was considering whether actions and discussion occur together, but not the nature of the actions and the discussion that occur together [15]. The other approach analysed the synchronicity of actions and the transactivity of the discussions separately [23]. In this work, we bring together these two approaches and propose a combined differential sequence mining and qualitative content analysis approach to examine the transactivity of discussion and actions. Specifically, we ask the following questions:

- RQ1: What is the relationship between the discussion and actions, and how is this relationship different between gainers and non-gainers?
- RQ2: What is the qualitative nature of verbal interactions that happen along with a specific action of interest?

We begin with the data of 32 teams working on a collaborative robot-mediated problem-solving activity where *actions* refer to any interaction with the activity interface and *discussions* refer to quantity and quality of communication between the two team members. To answer RQ1 we performed differential sequence mining on the combined speech and action sequence to identify the relationship between actions and speech, which actions are accompanied by speech, and how this differs between gainers and non-gainers. Next, to answer RQ2 we perform content analysis of the discussion occurring around one particular action of interest and examine the nature of discussion and how it varies between gainers and non-gainers. Our two part approach helps us illustrate the notion of action-discussion transactivity that is conducive to learning and we find that reflective actions accompanied with elaboration, reflection, negotiation and planning regarding next steps, are related with learning. The main contributions of this work are the notion of action-discussion transactivity and a methodology to examine the productivity of collaborative learning with this lens.

2. RELATED WORK

Research on collaborative learning has shown the key role of verbal interaction in advancing thinking and learning [26, 3]. Groups that are successful in problem-solving usually discuss and accept the correct proposal and their discussions are more coherent [3]. Conversation is the process by which students build and maintain a joint problem space [24]. Transactive verbal interaction, which is characterized by partner's building on each other's reasoning, can improve learning as peers can generate more complex understanding of the problem quickly through such verbal interactions [26]. When they generate explanations during collaboration peers construct shared representations and this may be one of the mechanisms that results in knowledge co-construction [12]. Actions done within a CSCL environment can also create shared representations, which can be then referred to during the discussion and thus improve the quality of collaboration [6, 2].

In this direction, research identified productive action patterns during collaborative learning with an interactive tabletop by analysing action logs with and without verbal interaction [7, 15, 23, 8]. [8] found that the number of touches allowed (single or multiple) on the table did not affect the level or symmetry of physical or verbal participation, but the nature of the discussion, which was more task-focussed in the multi-touch condition. [15] found that while the level or symmetry of participation of each team member in terms of action and speech did not relate to collaboration quality, certain sequences of actions and speech were related to the quality of collaboration. Concretely, more collaborative groups have more patterns of verbal discussion accompanied with actions, less concurrency of actions and less parallel actions. On the other hand, less collaborative groups had actions with limited verbal interactions, high concurrency and parallelism. This suggests that students in less collaborative groups were not as aware of their peers actions and did not discuss about the actions. On the other hand, in a chat-based collaborative learning environment, researchers found that neither synchrony of students actions nor transactivity of students' chats was related to performance on the task, but other factors such as group dynamics and prior knowledge had a more crucial role [23]. Thus, the role of symmetry, synchrony and transactivity of actions and discussion during collaborative learning appears to depend on the context.

In CSCL environments, several metrics of student dialogue (speech or chats) have been identified which are indicative of good collaboration. These include quantity (eg, number and length of utterances, and talk time) and heterogeneity and transactivity of verbal participation (eg, turn taking and building on each other's reasoning), along with features of speech such as voice inflection [29, 27]. Going further, research employed a combination of audio and action features to measure the quality of collaboration and collaborative learning and found that classifiers using a combination of audio and action features always perform better than those classifiers using audio or actions alone [27, 22]. This suggests that combining conversation and action metrics together can offer a better understanding of the quality of collaboration. In this work, we build on this line of research by specifically examining the role of action-discussion transactivity in collaborative learning, i.e, how actions and discussions can build on each other to lead to learning.

3. METHODS

3.1 Learning Activity and Dataset

In order to understand action-discussion transactivity we propose an approach which combines differential sequence mining and qualitative content analysis, and choosing one type of action - reflective actions - as an example, we show how our analysis can identify what type of actions and discussion occurring together can lead to learning. We use the speech and log action data from a multimodal temporal dataset [17], and log actions and transcripts from its corresponding dialogue corpus [20] collected from a robot mediated collaborative learning activity called JUSThink [19]. In JUSThink, two children play as a team to solve a minimum spanning tree problem where the goal is to build railway tracks to connect gold mines on a fictional Swiss map with a minimum cost, as shown in Figure 1. The corpus comprises data from 64 children aged 9 to 12 years, grouped into 32 teams, from international schools in Switzerland. The children were familiar with collaborative activities and robots as part of school activities, but did not have prior experience with QTrobot. The study was not part of a regular classroom activity. Two different views are provided in this activity, namely figurative and abstract as shown in Figure 2, and each child in a team only has one view at a time. In the figurative view, one can add or remove tracks while in the abstract view, one can see the cost associated with building a track and review the team's previous solutions. Thus at a time, one child can do solution building actions while the other can do reflective actions, so they have to discuss with each other to plan the next steps. The views of the team members are swapped every two edits. Hence, with these collaborative script choices such as partial information and role switching, only one member can perform an action at a time, therefore every action is a team action. Teams are allowed to submit solutions multiple times until the time limit runs out. They can also check descriptions of activity functionality and rules on the help page, which has been elaborated for them by the robot before the activity starts. More details of the activity can be found in [19].

3.2 Feature Selection and Encoding

The original multimodal temporal dataset consists of 56 features including log features, audio features, video features etc. Our analysis only focuses on the log features and speech features. Therefore, we selected 5 features from the multimodal temporal dataset including T_add, T_remove, T_hist, T_speech, T_overlap_over_speech. With 32 teams, we have a total of 4676 time windows in our analysis where each time window corresponds to 10 seconds of activity. For each time window, we have three descriptive features additionally, which are team number, time_in_secs, and window number. The log and speech features as well as the three descriptive features are shown in Table 1.

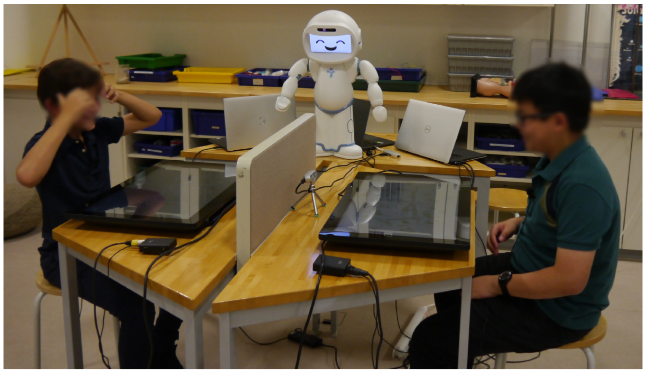


Figure 1: The experimental set up of JUSThink

Table 1: Multi-modal Features

Feature	Meaning
team	The team to which the window belongs to
time_in_secs	Time in seconds until that window
window	The window number
T_add	The number of times a team added an edge on the map
T_remove	The number of times a team removed an edge from the map
T_hist	The number of times a team opened the sub-window with history of their previous solutions
T_speech	The average of the two team member's speech activity in that window/(until that window)
T_overlap_over_speech	The average percentage of time the speech of the team members overlaps in that window/(until that window).

We begin by encoding the log and speech features in each 10s time window so that we can get a sequence representing the action+speech of each student in 10s increments. In the data set, the choice of 10 seconds as the unit of analysis is set considering the need to balance between too few and too many robot interventions. Before diving into the encoding details, we briefly elaborate on the terminology of gainers and non-gainers that we will be using from here onwards. In previous work based on this study [17], authors clustered the teams in two ways, once on the multimodal behaviors and once on task performance and learning gains (calculated as the normalized difference between pre and post test scores). Then, comparing the clusters using a similarity metric, they found higher learning gains associated with two sets of multimodal behaviors while lower learning gains were associated with another set of behaviors. They named the former set of 26 teams as gainers (those who gain knowledge) and the latter 6 teams as non-gainers (those who do not gain knowledge). For speech, it was found in [18] that speech behaviors are different for gainers and non-gainers, in terms of both quantity of speech and overlapping speech. So we define three levels - low, medium and high level of speech/speech overlap in each window on the basis of low and high thresholds of speech/speech overlap defined by considering the average of the 25th (for the low thresholds) and the 75th (for

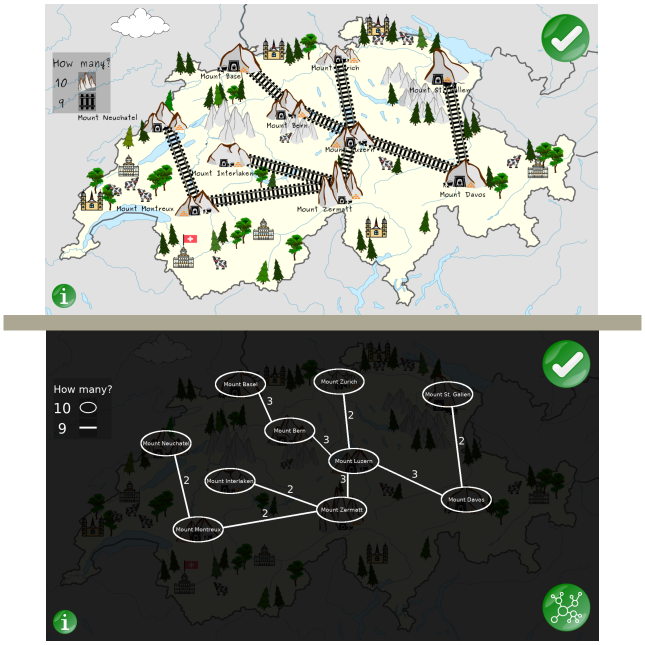


Figure 2: The interface for JUSThink where the two screens, separated by a barrier, show two different views: a figurative view (above) that allows for interactions such as additions, deletions of rail tracks, and an abstract view (below) that showcases the associated costs.

the high thresholds) percentiles across the gainers and non-gainers participants.Then we encode T_speech or T_overlap_over_speech in each window by using the low and high thresholds as shown in Table 2.

Table 2: Definition of Speech/Speech Overlap levels where x is the continuous value of Speech/Speech Overlap in a time window

speech/ speech overlap level	**condition**
LS/LSo	x <= low threshold
MS/MSo	low threshold <x <= high threshold
HS/HSo	x >high threshold

For action logs, we only consider add edges (T_add), remove edges (T_remove) and click solution history button (T_hist) within each time window as these are the meaningful actions that have been found to contribute to learning in this context[18]. We identify which of the meaningful actions happen in a time window and if an action happened at least once in a time window, it is encoded as being present. Then we have eight action combinations because each of the three actions can be either present or absent. Finally there are 24 combinations of action + speech in each time window (combinations of three levels of speech and eight action combinations) and we encode those combinations to 24 numbers as shown in the following Table 3. Note that the same encoding process is also applied to combinations of speech overlap levels and meaningful actions.

After encoding features in each time window of each team, we obtained two datasets of encoded features - one is encoded combinations of speech levels and actions, another is encoded combination of speech overlap levels and actions. Each dataset contains 32 teams' sequences of activities in ten-second time windows and we further separate each dataset into gainer sequences and non-gainer sequences for analysis.

3.3 Differential Sequence Mining

To answer our RQ1 by differentiating action+ speech sequences between gainers and non-gainers, we applied differential sequence mining algorithm (DSM)[13].DSM algorithm mainly uses the following two sequential pattern mining frequency measures.

1. *sequence support (s-support):* For a set of sequences, the number of sequences in which the pattern occurs, regardless of how frequently it occurs within each sequence.

2. *instance support (i-support):* For a given sequence, the number of times the pattern occurs, without overlap, in this sequence.

The algorithm firstly finds all patterns that meet the predefined s-support threshold. Then the algorithm selects only those patterns that have statistically significantly different i-support values between the two groups. Concretely, the algorithm filters frequent patterns based on the p-value of a t-test comparing the i-support values of patterns in each sequence, between the groups to find patterns whose p-value is less than 0.05. Finally, the algorithm compares the mean i-support value for each pattern between groups to identify the patterns that occur more often in one group than the other.

Before applying the DSM, we separated the two datasets we get after feature selection and encoding into four datasets. For each of the original two datasets as described in the previous sub-section, we divide the dataset (which contain 32 teams in total) into a sub-dataset that contained sequences of 26 gainer teams and another sub-dataset which contained sequences of 6 non-gainer teams.

Firstly, we set the minimum threshold of s-support to 0.6 and consider patterns that occur in at least 60% of sequences as s-frequent patterns within a group. We employ a simple sequential mining algorithm SPAMc [9] to find frequent patterns for both gainers and non-gainers with the LASAT tool [16]. Then we calculate the i-support of each frequent pattern in each team sequence in both gainers and non-gainers. For each frequent pattern, we generate a vector that contains i-support for each team sequence. Then we apply Welch's t-test with 0.05 p-value threshold to filter frequent patterns that are significantly different between gainers and non-gainers. After the filtering, we compare the mean i-support value for each frequent pattern between gainers and non-gainers so that we could compare patterns that occur more often in one group than the other. Finally, we get four categories of frequent patterns - two categories in which the patterns are s-frequent in only one group, and two categories in which the patterns are frequent in both groups but occurred more often in one group than the other.

Table 3: Encoding at specific level

Speech Level Code	Speech Overlap Level Code	Meaning
LS_Add	LSo_Add	Low level of speech(S)/speech overlap (So), at least add one edge
MS_Add	MSo_Add	Medium level of speech(S)/speech overlap (So), at least add one edge
HS_Add	HSo_Add	High level of speech(S)/speech overlap (So), at least add one edge
LS_Remove	LSo_Remove	Low level of speech(S)/speech overlap (So), at least remove one edge
MS_Remove	MSo_Remove	Medium level of speech(S)/speech overlap (So), at least remove one edge
HS_Remove	HSo_Remove	High level of speech(S)/speech overlap (So), at least remove one edge
LS_Hist	LSo_Hist	Low level of speech(S)/speech overlap (So) and at least click history button one time
MS_Hist	MSo_Hist	Medium level of speech(S)/speech overlap (So) and at least click history button one time
HS_Hist	HSo_Hist	High level of speech(S)/speech overlap (So) and at least click history button one time
LS_Add_Remove	LSo_Add_Remove	Low level of speech(S)/speech overlap (So), at least add one edge, at least remove one edge
MS_Add_Remove	MSo_Add_Remove	Medium level of speech(S)/speech overlap (So), at least add one edge, at least remove one edge
HS_Add_Remove	HSo_Add_Remove	High level of speech(S)/speech overlap (So),at least add one edge, at least remove one edge
LS_Add_Remove_Hist	LSo_Add_Remove_Hist	Low level of speech(S)/speech overlap (So), at least add one edge, at least remove one edge and at least click history button one time
MS_Add_Remove_Hist	MSo_Add_Remove_Hist	Medium level of speech(S)/speech overlap (So),at least add one edge, at least remove one edge and at least click history button one time
HS_Add_Remove_Hist	HSo_Add_Remove_Hist	High level of speech(S)/speech overlap (So), at least add one edge, at least remove one edge and at least click history button one time
LS_Remove_Hist	LSo_Remove_Hist	Low level of speech(S)/speech overlap (So), at least remove one edge and at least click history button one time
MS_Remove_Hist	MSo_Remove_Hist	Medium level of speech(S)/speech overlap (So), at least remove one edge and at least click history button one time
HS_Remove_Hist	HSo_Remove_Hist	High level of speech(S)/speech overlap (So), at least remove one edge and at least click history button one time
LS_Add_Hist	LSo_Add_Hist	Low level of speech(S)/speech overlap (So), at least add one edge and at least click history button one time
MS_Add_Hist	MSo_Add_Hist	Medium level of speech(S)/speech overlap (So), at least add one edge and at least click history button one time
HS_Add_Hist	HSo_Add_Hist	High level of speech(S)/speech overlap (So), at least add one edge and at least click history button one time
LS_NA	LSo_NA	Low level of speech(S)/speech overlap (So) and no useful action happens
MS_NA	MSo_NA	Medium level of speech(S)/speech overlap (So) and no useful action happens
HS_NA	HSo_NA	High level of speech(S)/speech overlap (So) and no useful action happens

3.4 Identifying relevant episodes of interest

To gain more insights into the transactivity of speech and log actions, we need to begin by identifying "patterns of interest" within the frequent patterns. From literature and previous research on the same dataset[18], we know that reflective speech and actions differentiate between gainers and non-gainers. Therefore, we are interested in reflection related actions (particularly clicking history button) and want to find what kind of verbal interaction happened along with it. So we consider frequent patterns which have speech along with open history action as episodes of interest for further qualitative analysis.

To find the exact content of dialogues in the relevant episodes, we matched the time window of the relevant episodes of each team to their transcript datasets in JUSThink Dialogue and Actions Corpus[20]. Due to the imprecision in matching time windows and the fact that it is difficult to extract meaningful information from very short segments (less than 60 seconds) of the dialogue, we have also included the conversation within 20 seconds before and after the matching time window.

3.5 Content Analysis on Dialogues

To answer RQ2 and examine the qualitative nature of dialogue during the episodes of interest, we decided to perform content analysis [14] on the selected dialogues. Content analysis is a qualitative analysis approach to code a corpus of data according to certain existing categories with the goal of doing statistical analysis on the numbers and identifying certain trends or providing evidence for/against a hypothesis. To look deeper into reflection behaviours, we focus on the following three aspects of problem-solving discussions and code the dialogue for these aspects:

1. What do teams observe from past actions?
2. What decisions do they take about future actions on the basis of these observations?
3. Do they reach any agreement on the future actions, and if yes, how?

In order to code their negotiation and agreement on their future actions, we applied the "refine" strategy in the negotiation framework[1] to analyse all dialogues. The "refine" strategy means that an agent decides to make another offer that somehow "refines", "builds" or "modifies" the original offer proposed by another agent. In this strategy, the initiating move includes an offer which is proposed by a speaker for agreement. The reactive moves include acceptance, ratification and rejection. Ratification refers to an acceptance which follows an acceptance by the other. Additionally, after an initial coding of the data, we defined additional categories in the negotiation framework, i.e., "goal clarification" and "sharing understanding" because they are relevant for this collaborative educational setting. The goal of this activity is to build tracks with the minimum cost – 22 francs. Team members must clarify this goal and share their understanding of the problem with each other due to the fact that they have two complementary views of the problem.

To illustrate the content analysis we conduct, we show some representative dialogues from both gainers and non-gainers in Table 12. For gainers' dialogue with index 1, team 8 set a wrong goal that they need to achieve the cost of 34 after the submission since they do not seem to understand the meaning of the word minimum. They make a decision to start in the middle and go around it as future problem solving steps. Then team 8 correct their previous wrong goal clarification in the dialogue with index 4 and decide to find the route that costs a lot as they perhaps want to remove the route with high cost.

4. RESULTS

4.1 *RQ1: What is the relationship between the discussion and actions, and how is this relationship different between gainers and non-gainers??*

The results of the DSM between the gainers and non-gainers action and speech sequences show that there are 12 patterns that are only frequent among gainers as shown in Table 4 and 17 patterns that are only frequent among non-gainers as shown in Table 5. Four patterns are frequent among both gainers and non-gainers, but occur more often among gainers as shown in Table 6 and five patterns are frequent among both gainers and non-gainers, but occur more often among non-gainers as shown in Table 7. Note that in the interest of the space, where there are several patterns we report only the top-10 patterns here and the full list is available in the appendix. In the following we elaborate on the obtained patterns; while there were several interesting patterns, we focus only on patterns which contain speech and actions together as our interest is on action-discussion transactivity.

As shown in Table 4, 92% (11/12) frequent patterns of gainers start with high speech, and 45% (5/11) of them are actions of adding edges with high speech level. The mean i-support of HS_Hist (clicking on the review history button with high level of speech) of gainers is 4.00 which is almost 6 times as much as that of non-gainers (0.67). This suggests that compared with non-gainers, gainers tend to review history along with long periods of discussion more frequently. Further, the mean i-support of HS_Remove (remove an edge with high levels of speech) of gainers is 3.54 which is more than twice that of non-gainers (1.50). This indicates that gainers remove an edge (a reflection action) along with high discussion more frequently than non-gainers.

For speech level related patterns that are frequent only among non-gainers as shown in Table 5 or more frequent among non-gainers (Table 7), barring 2 patterns in all the other patterns either the action of adding an edge or no action happens along with low level of speech. This suggests that compared with gainers, non-gainers tend to add edges more frequently and don't do as many reflection related actions (click review history button and remove edge) frequently, and that their actions are accompanied by low/medium levels of speech.

For the DSM of speech overlap level sequences, there are 16 patterns that are only frequent among gainers (Table 8) and 25 patterns that are only frequent among non-gainers (Table 9). Besides, seven patterns are frequent among both gainers and non-gainers, but occur more often among gainers as shown in Table 10 and four patterns are frequent among both gainers and non-gainers, but occur more often among non-gainers as shown in Table 11.

From the patterns that are only frequent among gainers (Table 8), we see that the mean i-support of the pattern high level of speech overlap while clicking review history button (HSo_Hist) among gainers (4.08) is more than twice the mean i-support among non-gainers (1.5). This pattern indicates that gainers have high level of speech overlap (interjecting speech) with the action of clicking the review history button. High level of speech overlap along with removing an edge (HSo_Remove) is also frequent only among gainers. This indicates that gainers more frequently have high level of overlapping speech while doing reflective actions such as reviewing history or removing an edge.

There is however one frequent pattern with reflective behaviours seen only among non-gainers in Table 15: medium level of speech overlap with clicking review history button, followed by low level of speech overlap without any meaningful action ([MSo Hist, LSo NA]). Compared with the pattern [HSo Hist, HSo NA] (see Table 14) that is only frequent among gainers, the difference is the level of speech overlap. We may infer that because non-gainers communicate less when they click the review history button, they perhaps take away less information from the history (reflection) than gainers and we examine this in depth in the next section.

To summarize the above findings, gainers perform more reflection related actions (review history, remove edges) along with higher level of speech/speech overlap compared with non-gainers. Therefore, gainers reflected more via discussion and improved their solutions based on previous solutions continuously.

Table 4: Patterns related to speech level and meaningful actions frequent only in gainers

frequent_pattern	p_value	mean_gainer_i_support	mean_non_gainer_i_support	diff_mean_i_support
['HS_NA', 'HS_NA']	0.003	9.96	1.67	8.3
['HS_Add', 'HS_Add']	3.4E-04	5.69	0.83	4.86
['HS_NA', 'HS_Add']	1.960E-04	5.19	0.5	4.69
['HS_Add', 'HS_NA']	1.6E-04	4.39	0.67	3.72
['HS_NA', 'HS_NA', 'HS_NA']	0.017	4.0	0.67	3.3
['HS_Hist']	3.4E-03	4.0	0.67	3.3
['HS_Remove']	0.048	3.54	1.5	2.03
['HS_Add', 'MS_Add']	1.7E-04	3.23	0.67	2.56
['MS_Add', 'HS_Add']	1.7E-03	3.1	0.67	2.41
['HS_NA', 'HS_NA', 'MS_NA']	8.5E-03	1.73	0.33	1.4
['HS_Add_Hist']	0.028	1.69	0.5	1.19
['HS_Add', 'HS_Add', 'MS_Add']	7.7E-05	0.96	0.0	0.96

Table 5: Top 10 patterns related to speech level and meaningful action frequent only in non-gainers

frequent_pattern	p_value	mean_gainer_i_support	mean_non_gainer_i_support	diff_mean_i_support
['LS_Add', 'LS_NA']	5.441E-03	0.92	5.17	-4.25
['LS_Add', 'LS_Add']	1.173E-03	1.8	5.0	-3.2
['LS_NA', 'LS_Add']	1.341E-02	1.0	4.17	-3.17
['LS_Add', 'LS_NA', 'LS_NA']	4.124E-02	0.19	2.5	-2.31
['LS_NA', 'LS_NA', 'MS_NA']	2.759E-02	0.46	2.33	-1.87
['LS_Add', 'LS_Add', 'LS_Add']	9.626E-03	0.62	2.17	-1.55
['LS_NA', 'MS_Add']	1.513E-02	0.73	2.0	-1.27
['LS_NA', 'LS_NA', 'LS_Add']	2.554E-02	0.35	1.83	-1.48
['LS_NA', 'LS_Add', 'LS_NA']	3.035E-02	0.23	1.5	-1.27
['LS_NA', 'LS_NA', 'LS_NA', 'LS_Add']	1.418E-02	0.27	1.5	-1.23

4.1.1 *Discussions while performing actions or when pausing actions*

We are specifically interested in whether actions are performed with speech or whether actions are paused during speech and the difference between gainers and non-gainers in this regard. So, we calculate the percentage of no useful action (NA) happening among frequent patterns in each category we get from DSM. For speech level related patterns, 38.36% of patterns that are only frequent among gainers do not have any meaningful action (NA) and 30.00% that are more frequent among gainers do not have any useful action (NA). On the other hand, 57.28% of patterns that are only frequent among non-gainers are without any useful action (NA) and 61.54% that are more frequent among non-gainers are without any useful action (NA). For speech overlap level related patterns, 33.62% of patterns that are only frequent among gainers do not have any useful action (NA) and 36.00% that are more frequent among gainers do not have any useful action (NA). While 54.24% of patterns that are only frequent among non-gainers are without any useful action (NA) and 57.89% that are more frequent among non-gainers are without any useful action (NA). These results indicate that speech and action occuring concurrently is more frequent among gainers than non-gainers.

4.2 *RQ2: What is qualitative nature of speech that occurs along with actions of interest?*

4.2.1 *Identifying relevant episodes of interests*

In [18], it was suggested that 1) speech overlap was one of behaviours which discriminated gainers from non-gainers in this context 2) students dialogue during episodes of speech overlap helped them build an understanding towards a solution and 3) gainers had more reflective actions than non-gainers. Therefore to explore the difference in the nature of the speech that occurs with actions, we choose reflective actions (specifically, reviewing history) as our action of interest. In our analysis above, we have identified some frequent, history related patterns that can serve as the relevant episodes of interest to perform the content analysis. Among those patterns, we pick [MSo_Hist, LSo_NA] which is only frequent among non-gainers and [HSo_Hist, HSo_NA] which is only frequent among gainers as the relevant episodes of interest for non-gainers and gainers, respectively (see appendix). In the current analysis, we focus on specific instances of when such speech overlap behaviors occur in conjunction with a action of reviewing history. The aforementioned two frequent patterns have the same time window length and similar actions but with different levels of speech overlap. An in-depth analysis of the content of dialogues happening during and around those episodes can help us better understand reflection behaviours of gainers and non-gainers, and any difference between them.

4.2.2 *Content Analysis of Dialogues*

We only have dialogue transcripts for a subset of teams (10) in JUSThink Dialogue and Actions Corpus[20]. After matching transcripts with relevant episodes, we get 13 dialogues from four gainer teams (teams 7, 8 , 9, and 47) and 4 dialogues from two non-gainer teams (teams 18, 20). Out of these 17 dialogues, 4 dialogues (24%) were analysed together by the first three authors of the paper until there

Table 6: **Frequent patterns related to speech level and exact meaningful action among both groups, but occurring more often in gainers**

frequent_pattern	p_value	mean_gainer_i_support	mean_non_gainer_i_support	diff_mean_i_support
['HS_NA']	4.858E-03	2.9	1.1	1.8
['HS_Add']	1.475E-06	20	4.5	15.5
['HS_Add_Remove']	1.678E-02	3.5	1.17	2.33
['MS_NA', 'HS_Add']	1.138E-02	2.1	0.8	1.3

Table 7: **Frequent patterns related to speech level and exact meaningful action among both groups, but occurring more often in non-gainers**

frequent_pattern	p_value	mean_gainer_i_support	mean_non_gainer_i_support	diff_mean_i_support
['LS_NA']	1.735E-02	10.15	36.50	-26.35
['LS_Add']	4.617E-03	6.69	17.67	-10.98
['LS_NA', 'LS_NA']	4.460E-02	2.77	16.17	-13.4
['MS_Hist']	1.393E-02	3.19	5.33	-2.14
['MS_Hist', 'MS_NA']	7.960E-03	0.92	2.17	-1.25

was complete agreement on the coding scheme. After these the remaining dialogues were analysed by one of the three researchers.

The codes for two exemplar gainer and non-gainer team dialogues are shown in Table 12. We began the analysis by summarizing the content of the dialogue. This was followed by coding for the negotiation mechanisms based on Baker's model of negotiation [1]. Finally, we coded for specific instances of reflection on past actions, planning for future actions and agreement because it is known that these shared regulation processes are necessary for collaborative problem-solving and learning [10].

The difference between the gainer and non-gainer teams is seen from their dialogues in Table 12. For instance, from the dialogues of non-gainer team 18 we see that each team member talks less compared with gainer team's dialogues. In dialogue with index 0, team 18 compares their previous solution with the current solution, but only speaker A performs some reflection and no one proposes any further steps to solve the problem. In the dialogue with index 1, team 18 discusses the result they get from their submission and decide to start over directly. Speaker A is still the only one who proposes ideas and B just follows A's requests. Further speaker A does not give any reason why he/she proposes those routes.

On the other hand in gainer team 8, two team members talk about their ideas actively. They always reflect on the previously submitted solution and clearly state a current problem solving strategy. For non-gainer team 18, they do not share their ideas as only one team member talks about his/her idea, and they reflect minimally on their previous solution. Compared with gainer team 8, non-gainer team 18 does not specify any further step to take in any episode.

Apart from these four representative dialogues, we conducted content analysis on all the available dialogues around relevant episodes for gainers (13 in total) and for non-gainers (4 in total). We found that for gainers' dialogues around relevant episodes, 58.3% (7/12) of them contain goal clarifications. Apart from dialogues around episodes 10 and 11, all the other dialogues -83.3% (10/12) dialogues - show some reflections from the past solutions. 91.7% (11/12) dialogues include making some decisions to take further steps based on past solutions. Offer-Acceptance happens more than twice in nearly half (5/12) of the dialogues.

In contrast, 25% (1/4) non-gainers' dialogues include goal clarifications. Only one non-gainer team takes some reflections from the previous solution and it is a wrong reflection. Only half of the dialogues contain decisions to take some steps for the future. There is no episode where offer-acceptance happens more than twice among non-gainers.

Our results are limited because of a skew in terms of much fewer numbers of non-gainer team dialogues than gainer team dialogues. Still, to summarize the findings of our content analysis, we note that gainers on average have more productive communication along with actions, because approximately half of their dialogues reached more than two agreements within 60 seconds as compared to none of the non-gainer dialogues. Gainers also tend to reflect more on past solutions and make timely decisions for future actions as compared to non-gainers.

5. DISCUSSION AND CONCLUSION

In this paper, we investigate the relationship between speech and actions, as well as the qualitative nature of the speech that occurs with the actions. Our first RQ was related to identifying the relationship between speech and actions. To answer this RQ, we applied differential sequence mining (DSM) to differentiate frequent patterns between gainers and non-gainers. We found that gainers and non-gainers demonstrate different relationships between speech and actions. Gainers perform all types of actions (solution building and reflective) along with high levels of speech/overlapping speech more frequently than non-gainers. While previous research indicated that gainers speak more [18], our findings nuance those findings by suggesting that speaking *while* performing actions is productive for learning. Our findings align with previous findings related to the concurrency of actions and speech among more collaborative groups [15]; however our findings extend to groups which learned more and are in a different collaborative scripted context. Gainers also show

Table 8: Top 10 patterns related to speech overlap level and exact meaningful action frequent only in gainers

frequent_pattern	p_value	mean_gainer_i_support	mean_non_gainer_i_support	diff_mean_i_support
['HSo_Add', 'HSo_Add']	7.9e-05	6.7	0.5	6.2
['HSo_NA', 'HSo_NA', 'HSo_NA']	0.006	5.3	1.0	4.3
['HSo_NA', 'HSo_Add']	1.1e-05	4.58	0.3	4.2
['HSo_Hist']	0.04	4.08	1.5	2.58
['HSo_Add', 'MSo_Add']	0.002	3.08	0.5	2.58
['HSo_Add', 'MSo_NA']	9.09e-09	2.73	0.17	2.56
['HSo_NA', 'MSo_Add']	0.0005	2.66	0.67	1.99
['MSo_NA', 'HSo_Add']	3.08e-06	2.66	0.33	2.32
['HSo_Add_Hist']	5.28e-05	2.15	0.0	2.15
['HSo_Add', 'HSo_NA', 'HSo_NA']	0.0004	1.46	0.0	1.46

Table 9: Top 10 patterns related to speech overlap level and exact meaningful action frequent only in non-gainers

frequent_pattern	p_value	mean_gainer_i_support	mean_non_gainer_i_support	diff_mean_i_support
['LSo_NA', 'LSo_NA', 'LSo_NA']	0.04	1.07	6.66	-5.58
['LSo_Add', 'LSo_Add']	0.00	1.34	5.83	-4.48
['LSo_Add', 'LSo_NA']	0.00	0.65	5.66	-5.01
['LSo_Remove']	0.03	1.0	4.5	-3.5
['LSo_NA', 'LSo_Add']	0.01	0.84	4.5	-3.65
['LSo_NA', 'MSo_Add']	0.00	0.65	3.33	-2.67
['LSo_NA', 'LSo_NA', 'LSo_NA', 'LSo_NA']	0.03	0.5	3.33	-2.83
['MSo_NA', 'LSo_Add']	0.00	0.88	2.83	-1.94
['MSo_Add', 'LSo_NA']	0.01	0.69	2.66	-1.97
['LSo_Add', 'LSo_NA', 'LSo_NA']	0.02	0.19	2.33	-2.14

longer patterns of continuous speech indicating that they communicate more actively since they usually have medium and high levels of speech/speech overlap while non-gainers often have low and medium levels of speech/speech overlap.

To look deeper into the speech that happens along with actions (RQ2), we performed content analysis on dialogues around the episodes of interest which are identified based on the DSM results. We already know that gainers tend to access the history (reflect) more [18], however here we additionally find that gainers share the information and understanding obtained from the reflective actions to a greater degree as they review their history with higher level of speech or speech overlap compared with non-gainers, and decide on future steps towards the goal. Perhaps non-gainers are unable to extract the needed information from their past solutions i.e, their reflective actions, which is why they do not discuss as much during the episodes of interest and do not arrive at a consensus regarding next steps. This suggests that some additional scaffolding is needed within the environment to point out to students what they should observe from the history. Another significant point of difference between gainers and non-gainers dialogue is that gainers clarify the goal of the task more frequently, which is a sign that the action of reviewing history is being used to ground their shared understanding of the task [2]. While DSM only shows us that some patterns are more frequent in one group vs the other, the qualitative analyses elaborate on how the patterns are different between the groups in terms of the content of their dialogue. The above findings, together with previous literature which suggests that elaborative discussions lead to learning in a collaborative scenario [26, 24], points to the fact that it is the nature of discussions during these differentiating patterns that could be reason for the difference in the learning between the groups.

Our presented approach combining DSM and qualitative analysis allows us to illustrate the importance of action-discussion transactivity in collaborative learning, and identify the nature of the discussion that can build on certain actions and make them productive. The qualitative analyses of the patterns unpacks the nature of the action-discussion patterns in each group. Specifically, we identify that the gainer groups do more elaborative, reflective and planning discussions which build on the history check action, compared to the non-gainer group. In other words, gainers had a greater degree of action-discussion transactivity, because they articulated their ideas and information obtained from doing the history check action, which helped them progress in the solution building. Compared to previous work [15, 23], our findings highlight the nature of the actions and discussions occurring together, how they build on each other and their association with collaborative learning, as opposed to the quality of collaboration or task performance. To summarize, our findings suggest that those who learned had a greater degree of action-discussion transactivity, and that they more frequently articulated their ideas and information obtained from doing actions, which helped them progress in the solution building. Taken together with previous literature in collaborative learning (eg. [21]) which speaks about the necessity of elaboration and transactivity in discussions and actions, our findings indicate that students should be encouraged to articulate their ideas or information obtained from doing actions and a teacher or a scaffold build into the CSCL environment can prompt the students to do so. Regulation of their performance and learning is challenging for students and researchers have proposed technological tools to support students [11]. This work provides sugges-

Table 10: Patterns related to speech overlap level and exact meaningful action frequent among both groups, but occurring more often in gainers

frequent_pattern	p_value	mean_gainer_i_support	mean_non_gainer_i_support	diff_mean_i_support
['HSo_NA']	0.00	31.69	12.66	19.02
['HSo_Add']	2.41e-08	21.53	3.0	18.53
['HSo_NA', 'HSo_NA']	0.00	11.5	3.0	8.5
['HSo_Remove']	0.00	4.38	1.33	3.05
['HSo_Add', 'HSo_NA']	7.17e-06	4.26	0.66	3.60
['HSo_Add_Remove']	0.00	4.0	1.0	3.0
['MSo_Add', 'HSo_Add']	0.00	3.0	0.83	2.16

Table 11: Patterns related to speech overlap level and exact meaningful actions among both groups, but occurring more often in non-gainers

frequent_pattern	p_value	mean_gainer_i_support	mean_non_gainer_i_support	diff_mean_i_support
['LSo_NA']	0.00	9.0	37.33	-28.33
['LSo_Add']	0.00	6.26	21.0	-14.73
['LSo_NA', 'LSo_NA']	0.03	2.57	15.0	-12.42
['MSo_NA', 'LSo_NA']	0.04	2.5	7.16	-4.66

tions regarding when such tools can be most productive for students, for instance, prompts for goal clarification after a failed problem-solving attempt.

Our analysis in this paper is limited from the following aspects. The number of non-gainer teams is much lesser than gainer teams, and as a result the number of non-gainer team dialogues is also much lesser than gainer team dialogues. However, the imbalance between gainers and non-gainers is due to the nature of the experiment setting - the experiment was designed to facilitate learning. Secondly, the interaction for each team of around 20-25 minutes is organized in windows of 10 seconds in the multimodal temporal dataset while dialogues start and end time are recorded as exact timestamps in the transcripts dataset. When we pick dialogues within identified relevant episodes, the difference in time features of the two datasets can cause slight inaccuracies of the matched results. To solve this problem, we pick up dialogues before and after 20 seconds of the relevant episode. Finally, the transcripts dataset does not include all teams. We have only analysed a subset of the dialogues from available transcripts. Our future work focuses on obtaining more data to extend this approach to larger sets of gainers and non-gainers, and other actions of interest in collaborative learning.

Table 12: Representative Dialogues

	index	team	dialogues	negotiation mechanism	general summary	reflect _past _act	reflect_ future _act	agreement
Gainers	1	8	A: ”it’s expensive you just used 5 .” B: ’go , to mount gallen .’ A: ”and i think we’re done .” B: ’go .’ A: ’i think we have more .’ B: ’wait !’ A: ”we’re done .” R: ’you are not that far from the minimum the difference is only 6 francs i am sure you can do it .’ A: ’can i show you how to do it ?’ B: ’oh , i know !’ ’wait .’ ’27 29 30 31 32 33 34 .’ A: ’what if we start in the middle and then go around it ?’] B: ’wait .’ ’34 .’ ’the minimum is 34 .’ B: ’see we have to spend okay .’ B: ’so do the circle , okay but do the circle , go .’ A: ’okay um .’ B: ’mount , neuchatel , okay , over there .’ B: ’then you have to make an’	A: Offer B:Acceptance for the immediate action but not for the quality of the solution A: Offer B: Acceptance A: Ratification	After the submission, they set a wrong goal to achieve 34. It seems that they misunderstand the meaning of minimum.	Wrong Goal Clarification - get the minimum: 34	start in the middle and then go around it	Yes for immediate action No for quality of the solution Yes for problem solving approach
	4	8	B: ”i’m gonna submit .” A: ’waiting for your’ R: ’you are not that far from the minimum the difference is only 7 francs i am sure you can do it’ A: ’7 francs.’ B: ’what ?’ ’what ? A: ’the minimum is 7 francs .’ B: ’uh we have to get 7 francs less .’ B: ’we know mount zermatt to mount interlaken is 4 francs and we know mount neuchatel to mount basel is 4 francs.’ B: ”we don’t want” A: ’one , one of them you said was 5. B: ’which , what ?’ A: ”i can’t remember which one that was though .” A: ’i think it was mount basel to mount zurich .’ B: ’no that was not never connected let me see . B: ’uh the one that was 5 was neuchatel to bern.’ A: ’yeah .”it’s hard’ B: ’um’	B: Offer A: Acceptance for the immediate action but not for the quality of the solution Share Understanding	Team 8 correct their previous misunderstand here.	Goal Clarification: they should spend 7 francs less	Find the route cost 5 frans	Yes for the immediate action but not for the quality of the solution
Non-gainers	0	18	B: ’oh yes , click the check .’ A: ’okay .’ R:’...’ B,:’what ?’ ’what’ ’oh now we start again , basically .’ A: ’oh , oh i see , compare solutions .’ B: ’what can we do ?’ A: ’oh okay um .’ A: ’oh this is our previous solution price 64 .’ B: ’so’ A: ’oh i get it we have to get the most price , but .’ I: ’....’ A: ’oh by by 40 francs so’ A: ’oh so we need to get 24 .’	B: Offer A: Acceptance A: Offer B: Acceptance A: Ratification	They reviewed their previous solution and they set the wrong goal that the minimum is 24.	Wrong Goal Clarification: ’we need to get 24’	No	Yes for immediate actions
	1	18	B: ’i think’ A: ’oh .’ B: ’so we submit it ?’ A: ”yeah let’s start over .” B: ’robot say something to us .’ R: ’...’ B: ’yeah .’ ’should we’ I: ’...’ A: ’oh .’ R: ”i don’t care we .” A: ’where is 2 ? ’give me a 2 .’ A’, ’2 francs one .’ B: ’2 francs um’ ’interlaken to zermatt .’ A: ’interlaken to zermatt . A: ’zermatt oh interlaken .’ ”that’s 2 ?” B: ’interlaken to zermatt .’ A: ”that’s 2 ?” B: ’yes .’ A: ’oh .’ B: ’and you want another 2 ?’ A: ’yeah .’ B: ’or 3 ?’ A: ’yeah uh as much 2s and then as much 3s .’ B: ’zermatt to montreux .’ ’montreux .’ A: ’oh montreux .’ ’uh’	B: Offer A: Acceptance Share Understanding Ask for something without sharing any idea	They tried to find those tracks with cost of 2 or 3 francs.	No	No	Yes for immediate actions

6. REFERENCES

[1] M. Baker. A model for negotiation in teaching-learning dialogues. *Journal of Interactive Learning Research*, 5(2):199, 1994.

[2] M. Baker, T. Hansen, R. Joiner, and D. Traum. The role of grounding in collaborative learning tasks. *Collaborative learning: Cognitive and computational approaches*, 31:63, 1999.

[3] B. Barron. When smart groups fail. *The journal of the learning sciences*, 12(3):307–359, 2003.

[4] J. Davidsen and T. Ryberg. "this is the size of one meter": Children's bodily-material collaboration. *International Journal of Computer-Supported Collaborative Learning*, 12(1):65–90, 2017.

[5] P. Dillenbourg, S. Järvelä, and F. Fischer. The evolution of research on computer-supported collaborative learning. In *Technology-enhanced learning*, pages 3–19. Springer, 2009.

[6] P. Dillenbourg and D. Traum. Sharing solutions: Persistence and grounding in multimodal collaborative problem solving. *The Journal of the Learning Sciences*, 15(1):121–151, 2006.

[7] A. C. Evans, J. O. Wobbrock, and K. Davis. Modeling collaboration patterns on an interactive tabletop in a classroom setting. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 860–871, 2016.

[8] A. Harris, J. Rick, V. Bonnett, N. Yuill, R. Fleck, P. Marshall, and Y. Rogers. Around the table: Are multiple-touch surfaces better than single-touch for children's collaborative interactions? In *CSCL (1)*, pages 335–344, 2009.

[9] J. Ho, L. Lukov, and S. Chawla. Sequential pattern mining with constraints on large protein databases. In *Proceedings of the 12th international conference on management of data (COMAD)*, pages 89–100, 2005.

[10] S. Järvelä and A. F. Hadwin. New frontiers: Regulating learning in cscl. *Educational psychologist*, 48(1):25–39, 2013.

[11] S. Järvelä, P. A. Kirschner, A. Hadwin, H. Järvenoja, J. Malmberg, M. Miller, and J. Laru. Socially shared regulation of learning in cscl: Understanding and prompting individual-and group-level shared regulatory activities. *International Journal of Computer-Supported Collaborative Learning*, 11(3):263–280, 2016.

[12] H. Jeong and M. T. Chi. Construction of shared knowledge during collaborative learning. 1997.

[13] J. S. Kinnebrew, K. M. Loretz, and G. Biswas. A contextualized, differential sequence mining method to derive students' learning behavior patterns. *Journal of Educational Data Mining*, 5(1):190–219, 2013.

[14] K. Krippendorff. Content analysis. 1989.

[15] R. Martinez-Maldonado, Y. Dimitriadis, A. Martinez-Monés, J. Kay, and K. Yacef. Capturing and analyzing verbal and physical collaborative learning interactions at an enriched interactive tabletop. *International Journal of Computer-Supported Collaborative Learning*, 8(4):455–485, 2013.

[16] S. Mishra, A. Munshi, M. Rushdy, and G. Biswas. Lasat: learning activity sequence analysis tool. In *Technology-enhanced & evidence-based education & learning (TEEL) workshop at the 9th international learning analytics and knowledge (LAK) conference, Tempe, Arizona, USA*, 2019.

[17] J. Nasir, B. Bruno, and P. Dillenbourg. PE-HRI-temporal: A Multimodal Temporal Dataset in a robot mediated Collaborative Educational Setting, Nov. 2021.

[18] J. Nasir, A. Kothiyal, B. Bruno, and P. Dillenbourg. Many are the ways to learn identifying multi-modal behavioral profiles of collaborative learning in constructivist activities. *International Journal of Computer-Supported Collaborative Learning*, 16(4):485–523, 2021.

[19] J. Nasir, U. Norman, B. Bruno, and P. Dillenbourg. When positive perception of the robot has no effect on learning. In *2020 29th IEEE international conference on robot and human interactive communication (RO-MAN)*, pages 313–320. IEEE, 2020.

[20] U. Norman, T. Dinkar, J. Nasir, B. Bruno, C. Clavel, and P. Dillenbourg. Justhink dialogue and actions corpus, Mar. 2021.

[21] O. Noroozi, A. Weinberger, H. Biemans, M. Mulder, and M. Chizari. Facilitating argumentative knowledge construction through a transactive discussion script in CSCL. 61, 2013.

[22] J. K. Olsen, K. Sharma, N. Rummel, and V. Aleven. Temporal analysis of multimodal data to predict collaborative learning outcomes. *British Journal of Educational Technology*, 51(5):1527–1547, 2020.

[23] V. Popov, A. van Leeuwen, and S. C. Buis. Are you with me or not? temporal synchronicity and transactivity during cscl. *Journal of Computer Assisted Learning*, 33(5):424–442, 2017.

[24] J. Roschelle and S. D. Teasley. The construction of shared knowledge in collaborative problem solving. In *Computer supported collaborative learning*, pages 69–97. Springer, 1995.

[25] J. R. Segedy, J. S. Kinnebrew, and G. Biswas. Using coherence analysis to characterize self-regulated learning behaviours in open-ended learning environments. *Journal of Learning Analytics*, 2(1):13–48, 2015.

[26] S. D. Teasley. The role of talk in children's peer collaborations. *Developmental psychology*, 31(2):207, 1995.

[27] S. A. Viswanathan and K. VanLehn. Using the tablet gestures and speech of pairs of students to classify their collaboration. *IEEE Transactions on Learning Technologies*, 11(2):230–242, 2017.

[28] F. Vogel, I. Kollar, S. Ufer, E. Reichersdorfer, K. Reiss, and F. Fischer. Developing argumentation skills in mathematics through computer-supported collaborative learning: The role of transactivity. *Instructional Science*, 44(5):477–500, 2016.

[29] A. Weinberger and F. Fischer. A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & education*, 46(1):71–95, 2006.

7. APPENDIX

Table 13: Speech level patterns frequent only among non-gainers

frequent_pattern	p_value	mean_gainer_i_support	mean_non_gainer_i_support	diff_mean_i_support	freq_decode_pattern
[0, 21]	5.441E-03	9.231E-01	5.167E+00	-4.244E+00	['LS_Add', 'LS_NA']
[0, 0]	1.173E-03	1.808E+00	5.000E+00	-3.192E+00	['LS_Add', 'LS_Add']
[21, 0]	1.341E-02	1.000E+00	4.167E+00	-3.167E+00	['LS_NA', 'LS_Add']
[0, 21, 21]	4.124E-02	1.923E-01	2.500E+00	-2.308E+00	['LS_Add', 'LS_NA', 'LS_NA']
[21, 21, 22]	2.759E-02	4.615E-01	2.333E+00	-1.872E+00	['LS_NA', 'LS_NA', 'MS_NA']
[0, 0, 0]	9.626E-03	6.154E-01	2.167E+00	-1.551E+00	['LS_Add', 'LS_Add', 'LS_Add']
[21, 1]	1.513E-02	7.308E-01	2.000E+00	-1.269E+00	['LS_NA', 'MS_Add']
[21, 21, 0]	2.554E-02	3.462E-01	1.833E+00	-1.487E+00	['LS_NA', 'LS_NA', 'LS_Add']
[21, 0, 21]	3.035E-02	2.308E-01	1.500E+00	-1.269E+00	['LS_NA', 'LS_Add', 'LS_NA']
[21, 21, 21, 0]	1.418E-02	2.692E-01	1.500E+00	-1.231E+00	['LS_NA', 'LS_NA', 'LS_NA', 'LS_Add']
[0, 0, 21]	2.575E-02	3.077E-01	1.333E+00	-1.026E+00	['LS_Add', 'LS_Add', 'LS_NA']
[21, 0, 0, 0]	2.913E-02	2.308E-01	1.000E+00	-7.692E-01	['LS_NA', 'LS_Add', 'LS_Add', 'LS_Add']
[21, 1, 22]	2.006E-02	1.538E-01	1.000E+00	-8.462E-01	['LS_NA', 'MS_Add', 'MS_NA']
[0, 0, 21, 21]	4.219E-02	0.000E+00	8.333E-01	-8.333E-01	['LS_Add', 'LS_Add', 'LS_NA', 'LS_NA']
[21, 22, 21, 22]	4.669E-02	1.154E-01	6.667E-01	-5.513E-01	['LS_NA', 'MS_NA', 'LS_NA', 'MS_NA']
[21, 22, 22, 22, 22]	4.669E-02	1.154E-01	6.667E-01	-5.513E-01	['LS_NA', 'MS_NA', 'MS_NA', 'MS_NA', 'MS_NA']
[0, 0, 0, 21]	3.716E-02	7.692E-02	6.667E-01	-5.897E-01	['LS_Add', 'LS_Add', 'LS_Add', 'LS_NA']

Table 14: Speech overlap level patterns only frequent among gainers

frequent_pattern	**p_value**	**mean_gainer_i_support**	**mean_non_gainer_i_support**	**diff_mean_i_support**	**freq_decode_pattern**
[2, 2]	7.928934765535499e-05	6.730769230769231	0.5	6.230769230769231	['HSo_Add', 'HSo_Add']
[23, 23, 23]	0.006002839704649775	5.3076923076923075	1.0	4.3076923076923075	['HSo_NA', 'HSo_NA', 'HSo_NA']
[23, 2]	1.1371592327013828e-05	4.576923076923077	0.3333333333333333	4.243589743589744	['HSo_NA', 'HSo_Add']
[8]	0.036533665925983824	4.076923076923077	1.5	2.5769230769230766	['HSo_Hist']
[2, 1]	0.0022205587982335228	3.076923076923077	0.5	2.576923076923077	['HSo_Add', 'MSo_Add']
[2, 22]	9.087126987994823e-09	2.730769230769231	0.16666666666666666	2.5641025641025643	['HSo_Add', 'MSo_NA']
[23, 1]	0.0005621716567395286	2.6538461538461537	0.6666666666666666	1.9871794871794872	['HSo_NA', 'MSo_Add']
[22, 2]	3.082910493793244e-06	2.6538461538461537	0.3333333333333333	2.3205128205128203	['MSo_NA', 'HSo_Add']
[20]	5.2819931032191796e-05	2.1538461538461537	0.0	2.1538461538461537	['HSo_Add_Hist']
[2, 23, 23]	0.00043067995679562895	1.4615384615384615	0.0	1.4615384615384615	['HSo_Add', 'HSo_NA', 'HSo_NA']
[23, 23, 2]	0.0019868197228214823	1.4615384615384615	0.0	1.4615384615384615	['HSo_NA', 'HSo_NA', 'HSo_Add']
[2, 2, 23]	0.0003332574321561345	1.4615384615384615	0.0	1.4615384615384615	['HSo_Add', 'HSo_Add', 'HSo_NA']
[23, 2, 2]	0.0006605272364293755	1.3076923076923077	0.0	1.3076923076923077	['HSo_NA', 'HSo_Add', 'HSo_Add']
[8, 23]	0.0004105089572129998	1.2692307692307692	0.0	1.2692307692307692	['HSo_Hist', 'HSo_NA']
[5, 23]	0.0030286337078061225	1.1538461538461537	0.16666666666666666	0.9871794871794871	['HSo_Remove', 'HSo_NA']
[22, 2, 2]	2.101257038827904e-05	0.8846153846153846	0.0	0.8846153846153846	['MSo_NA', 'HSo_Add', 'HSo_Add']

Table 15: Speech overlap level patterns only frequent among non-gainers

frequent_pattern	**p_value**	**mean_gainer_i_support**	**mean_non_gainer_i_support**	**diff_mean_i_support**	**freq_decode_pattern**
[21, 21, 21]	0.04939554474004129	1.0769230769230769	6.666666666666667	-5.58974358974359	['LSo_NA', 'LSo_NA', 'LSo_NA']
[0, 0]	0.0018586234032742692	1.3461538461538463	5.833333333333333	-4.487179487179487	['LSo_Add', 'LSo_Add']
[0, 21]	0.00016706039378552398	0.6538461538461539	5.666666666666667	-5.012820512820513	['LSo_Add', 'LSo_NA']
[3]	0.035109050581296	1.0	4.5	-3.5	['LSo_Remove']
[21, 0]	0.013026456729087523	0.8461538461538461	4.5	-3.6538461538461537	['LSo_NA', 'LSo_Add']
[21, 1]	0.008836545666620073	0.6538461538461539	3.3333333333333335	-2.6794871794871797	['LSo_NA', 'MSo_Add']
[21, 21, 21, 21]	0.03860139794859445	0.5	3.3333333333333335	-2.8333333333333335	['LSo_NA', 'LSo_NA', 'LSo_NA', 'LSo_NA']
[22, 0]	0.006750888971567344	0.8846153846153846	2.8333333333333335	-1.948717948717949	['MSo_NA', 'LSo_Add']
[1, 21]	0.01443329824984804	0.6923076923076923	2.6666666666666665	-1.9743589743589742	['MSo_Add', 'LSo_NA']
[0, 21, 21]	0.023209371486681146	0.19230769230769232	2.3333333333333335	-2.141025641025641	['LSo_Add', 'LSo_NA', 'LSo_NA']
[0, 0, 0]	0.009755427161716556	0.2692307692307692	2.0	-1.7307692307692308	['LSo_Add', 'LSo_Add', 'LSo_Add']
[0, 0, 21]	0.0056742454761588585	0.19230769230769232	1.6666666666666667	-1.4743589743589745	['LSo_Add', 'LSo_Add', 'LSo_NA']
[7, 21]	0.013914091446642511	0.11538461538461539	1.3333333333333333	-1.2179487179487178	['MSo_Hist', 'LSo_NA']
[21, 21, 1]	0.013914091446642511	0.11538461538461539	1.3333333333333333	-1.2179487179487178	['LSo_NA', 'LSo_NA', 'MSo_Add']
[21, 1, 21]	0.014075354800622239	0.038461538461538464	1.1666666666666667	-1.1282051282051282	['LSo_NA', 'MSo_Add', 'LSo_NA']
[0, 0, 21, 21]	0.011724811003954628	0.0	1.0	-1.0	['LSo_Add', 'LSo_Add', 'LSo_NA', 'LSo_NA']
[21, 0, 0, 0]	0.013173766481180184	0.038461538461538464	1.0	-0.9615384615384616	['LSo_NA', 'LSo_Add', 'LSo_Add', 'LSo_Add']
[0, 0, 0, 21]	0.04085940385929584	0.0	1.0	-1.0	['LSo_Add', 'LSo_Add', 'LSo_Add', 'LSo_NA']
[22, 22, 0]	0.020519815735647172	0.2692307692307692	0.8333333333333334	-0.5641025641025641	['MSo_NA', 'MSo_NA', 'LSo_Add']
[0, 21, 22]	0.0041047159800533225	0.0	0.8333333333333334	-0.8333333333333334	['LSo_Add', 'LSo_NA', 'MSo_NA']
[21, 21, 1, 21]	0.004307836785291385	0.038461538461538464	0.8333333333333334	-0.7948717948717949	['LSo_NA', 'LSo_NA', 'MSo_Add', 'LSo_NA']
[21, 21, 1, 21, 21]	0.02503101581845297	0.0	0.6666666666666666	-0.6666666666666666	['LSo_NA', 'LSo_NA', 'MSo_Add', 'LSo_NA', 'LSo_NA']
[21, 21, 22, 21]	0.04669295353054086	0.11538461538461539	0.6666666666666666	-0.5512820512820512	['LSo_NA', 'LSo_NA', 'MSo_NA', 'LSo_NA']
[21, 1, 21, 21]	0.02503101581845297	0.0	0.6666666666666666	-0.6666666666666666	['LSo_NA', 'MSo_Add', 'LSo_NA', 'LSo_NA']
[0, 21, 21, 21]	0.030127010101375896	0.038461538461538464	0.6666666666666666	-0.6282051282051282	['LSo_Add', 'LSo_NA', 'LSo_NA', 'LSo_NA']

Generalizing Predictive Models of Reading Ability in Adaptive Mathematics Software

Husni Almoubayyed
Carnegie Learning, Inc.
501 Grant St, Ste 1075
Pittsburgh, PA 15219
halmoubayyed@carnegielearning.com

Stephen E. Fancsali
Carnegie Learning, Inc.
501 Grant St, Ste 1075
Pittsburgh, PA 15219
sfancsali@carnegielearning.com

Steve Ritter
Carnegie Learning, Inc.
501 Grant St, Ste 1075
Pittsburgh, PA 15219
sritter@carnegielearning.org

ABSTRACT

Recent research seeks to develop more comprehensive learner models for adaptive learning software. For example, models of reading comprehension built using data from students' use of adaptive instructional software for mathematics have recently been developed. These models aim to deliver experiences that consider factors related to learning beyond performance in the target domain for instruction. We investigate the extent to which generalization is possible for a recently developed predictive model that seeks to infer students' reading comprehension ability (as measured by end-of-year standardized test scores) using an introductory learning experience in Carnegie Learning's MATHia intelligent tutoring system for mathematics. Building on a model learned on data from middle school students in a single school district in a mid-western U.S. state, using that state's end-of-year English Language Arts (ELA) standardized test score as an outcome, we consider data from a school district in a south-eastern U.S. state as well as that state's end-of-year ELA standardized test outcome. Generalization is explored by considering prediction performance when training and testing models on data from each of the individual school districts (and for their respective state's test outcomes) as well as pooling data from both districts together. We conclude with discussion of investigations of some algorithmic fairness characteristics of the learned models. The results suggest that a model trained on data from the smaller of the two school districts considered may achieve greater fairness in its predictions over models trained on data from the other district or both districts, despite broad, overall similarities in some demographic characteristics of the two school districts. This raises interesting questions for future research on generalizing these kinds of models as well as on ensuring algorithmic fairness of resulting models for use in real-world adaptive systems for learning.

H. Almoubayyed, S. Fancsali, and S. Ritter. Generalizing predictive models of reading ability in adaptive mathematics software. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 207–216, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115782

Keywords

student modeling, reading comprehension, intelligent tutoring systems, generalizability, algorithmic fairness, middle school mathematics, neural networks

1. INTRODUCTION

Recent research seeks to develop more comprehensive models of students using adaptive software for learning. Such models consider learning factors that are at least nominally beyond the scope of the learning software's target domain (e.g., modeling students' reading comprehension ability in the context of their usage of software for mathematics instruction) [15] [1]. Richey el al. [15] considered a particular piece of introductory instructional content in Carnegie Learning's MATHia (formerly Cognitive Tutor [16]) intelligent tutoring system (ITS) and used students' performance on that content as a proxy for their reading ability. Their argument for this choice was that performance measures for that content, generally providing instruction on how to use the ITS and its various support features, were more likely to be indicative of students' reading ability than their mathematics ability.

Almoubayyed et al. [1] built on this initial work by providing empirical support for the argument due to Richey and colleages [15], demonstrating that performance on this introductory MATHia content is correlated with students' performance on end-of-year standardized test scores for English Language Arts (ELA). Further, it was found that the correlation of student performance with ELA test scores compared to the correlation of student performance with mathematics test scores was greater than almost all other content in MATHia, suggesting the possibility that this early performance in MATHia might serve as a type of instruction-embedded assessment of reading ability. Such an assessment of reading ability, especially early within a student's use of MATHia or other adaptive software, might serve at least two purposes:

- Early prediction(s) that a student may still be emerging as a reader of English at their grade-level can serve as quick (relatively low-stakes) diagnoses that adaptive reading supports should be made available to students. In the situation in which such supports are broadly available to all users of software, then messaging prompts or similar "nudges" might be adaptively presented to suggest their usage to particular students based on these kinds of predictions.

- Predictions that a student is likely an emerging English language learner (ELL) or for some other reason is struggling to read can be used in retrospective analyses and design-loop adaptivity processes [Aleven et al., 2017] to better understand whether various software features, content improvements, and/or supports for reading, meta-cognition, or other learning factors are having their desired effect (e.g., via randomized experiments or so-called "A/B tests" [18]), especially if such features, content, or supports are targeting a particular population of learners like ELLs. In large-scale deployments of adaptive learning software like MATHia, standardized test outcome data or student-level characteristics like ELL status are generally not available, neither to the software at run-time, nor to developers and analysts who seek to better understand how to improve users' learning experiences.

Almoubayyed et al. [1] develop neural network based prediction models for ELA exam scores that use performance features in this introductory content that are promising for at least the two above uses-cases.[1] These models were trained and tested on data from hundreds of students, including data for hundreds of thousands of student actions, in a single school district in a mid-western U.S. state. A natural question concerns the extent to which models learned in a single school district (and state) generalize to other school districts in other states. We build on the work of Almoubayyed et al. [1] to consider this question of generalizability.

2. MATHIA

MATHia (formerly Cognitive Tutor [16]) is an ITS for mathematics instruction that is a part of a blended, basal curriculum for middle school and high school mathematics developed by Carnegie Learning, and used by around half a million students across the United States. Instruction in MATHia is delivered via complex, multi-step problems, with most steps within problems mapped to one or more knowledge components (KCs, or skills [11]). Students work through "workspaces" that provide practice on a set of KCs until the ITS has determined that the student has reached mastery [3] of all such KCs (using the Bayesian Knowledge Tracing framework [2]) in the workspace (or the student reaches a pre-defined maximum number of problems). When the student reaches mastery of all KCs (or the maximum number of problems), the student is moved on to the next workspace in the curriculum set by their teacher or school for their grade-level.

To introduce students to the practice opportunities they will receive in MATHia, the first workspace in MATHia, referred to by MATHia developers as the Pre-Launch Protocol, introduces students to the ITS software, its user-interface (e.g., how to watch videos and provide input to the ITS), adaptive support features like just-in-time (JIT) feedback and context-sensitive hints, as well as providing some motivational messaging about "growth-mind-set" [13] and related ideas (e.g., the video about "growing your brain" visible in the screenshot provided by Figure 1). Problems in the Pre-Launch Protocol are not necessarily about mathematics, but rather engage students with questions that are nearly certain to require students to engage with adaptive features of the software, such as hint requests. For example, one question asks students to provide the name of an animal that begins with the letter "e." Since the answer is not obvious (e.g., not "elephant"), students almost always have to request a hint and receive feedback on incorrect answers as they make attempts to correctly guess what the ITS is "thinking" about. The Pre-Launch Protocol is a non-mastery workspace in MATHia, and performance on the Pre-Launch Protocol is not related to KCs, but students' interactions, attempts, and correctness is nonetheless tracked in the Pre-Launch Protocol. Student performance data from the Pre-Launch Protocol workspace have figured prominently in two previous papers on developing more comprehensive models of reading comprehension while students use MATHia [15], [1].

The usefulness of the Pre-Launch Protocol in this context is due to several reasons: Firstly, the Pre-Launch Protocol is the very first thing that a student interacts with in MATHia, and therefore, the possibility of making accurate predictions using only Pre-Launch Protocol data can be powerful. Such predictions can be used to improve and personalize students' learning experiences in MATHia very early on in the academic year (whereas making a prediction near the end of the year would be less useful for many applications). Secondly, while content in intelligent tutoring systems is typically personalized to the student, and thus not every student encounters the same problems, that issue is not relevant for the Pre-Launch Protocol. Every student using MATHia completes an identical Pre-Launch Protocol, resulting in complete data. Finally, expecting the Pre-Launch Protocol to have predictive signal about factors of student learning not related to mathematics is well-motivated, due to the fact that it is the only piece of content in MATHia that does not deliver content directly related to mathematics or the student's curriculum.

The present work builds on the intuition of Richey et al. [15] and the initial empirical validation of their argument by Almoubayyed et al. [1] that student performance in this introductory content may serve as an instruction-embedded assessment of reading ability[2] that can be used to develop a more comprehensive student model within an ITS for mathematics. By considering additional data than these previous works, we seek to better understand whether the predictive

[1] Models developed by Almoubayyed et al. [1] that consider data from content beyond introductory content may be especially useful for retrospective analyses germane to the second use-case.

[2] Using statistical models of student performance and predictions about behavior and affective states in systems like MATHia as instruction-embedded assessments for the system's target domain (i.e., for predicting mathematics standardized test scores) has been explored in some depth across software platforms and U.S. states (e.g., [17] [12] [5]).

model developed in [1] generalizes to a new school district context. The new school district context includes a larger sample of students in a different U.S. state with outcome measures from a different standardized test. We now consider our data in more detail.

3. DATA

Relying partially on data provided by the authors of [1], we use two datasets of student end-of-year English Language Arts (ELA) standardized (state) test scores in Grade 7 in the 2021-2022 academic year. The datasets come from two school districts: one from a mid-western U.S. state that was studied in [1] and one from a south-eastern U.S. state. Hereafter, we refer to the dataset from the mid-western state as MW, the dataset from the south-eastern state as SE, and the combination of both as the Combined dataset. The datasets additionally include demographic information of the students. Although the demographics were similar in some aspects, for example, around 60% of the student population in both districts were white; there were large differences in overall student performance between them. Specifically, 78% of students in our MW dataset passed their end-of-year ELA state test, compared to 49% in our SE dataset. There were also a large difference between the size of the districts and the Grade 7 students for whom we have data, while MW had 831 students, SE had 4,349 students. For the purposes of this study, we categorized student performance as a binary measure of either passing or failing to pass the state test. We also received access to the students' action-level performance in MATHia on the 36-step Pre-Launch Protocol. In total, we received 563,650 action-level student records for the two districts combined, which is equivalent to 3 actions per step per student on average. There was no missing data for any student for any step: because the Pre-Launch Protocol is the first workspace a student interacts with in MATHia, and is presented identically across students, every student completed every step in the Pre-Launch Protocol. Students can either make an attempt or request a hint. If a student makes an incorrect attempt, they may receive JIT feedback if their mistake is deemed by MATHia as a "common misconception." Following the feature engineering steps that Almoubayyed, et al. defined in [1], we generate the following features from the data:

- `correct`: Whether a student's first attempt on a step was correct (1) or incorrect (0).
- `hint`: The number of hints that a student requested on a step. This number can be between 0 and 3.
- `jit`: The number of JIT feedback a student received on a step
- `attempt`: The number of attempts a student made on a step until reaching the correct answer.

We split each of the datasets into a training set and a test set, each containing half of the number of students selected at random. When training and testing on a combination of the datasets, we combine the two training set and the two test sets separately.

Test score and demographic data were provided by the two districts to Carnegie Learning according to data sharing agreements between Carnegie Learning and the district that allows for the use of these data for research purposes.

4. READING ABILITY PREDICTIVE MODEL GENERALIZATION

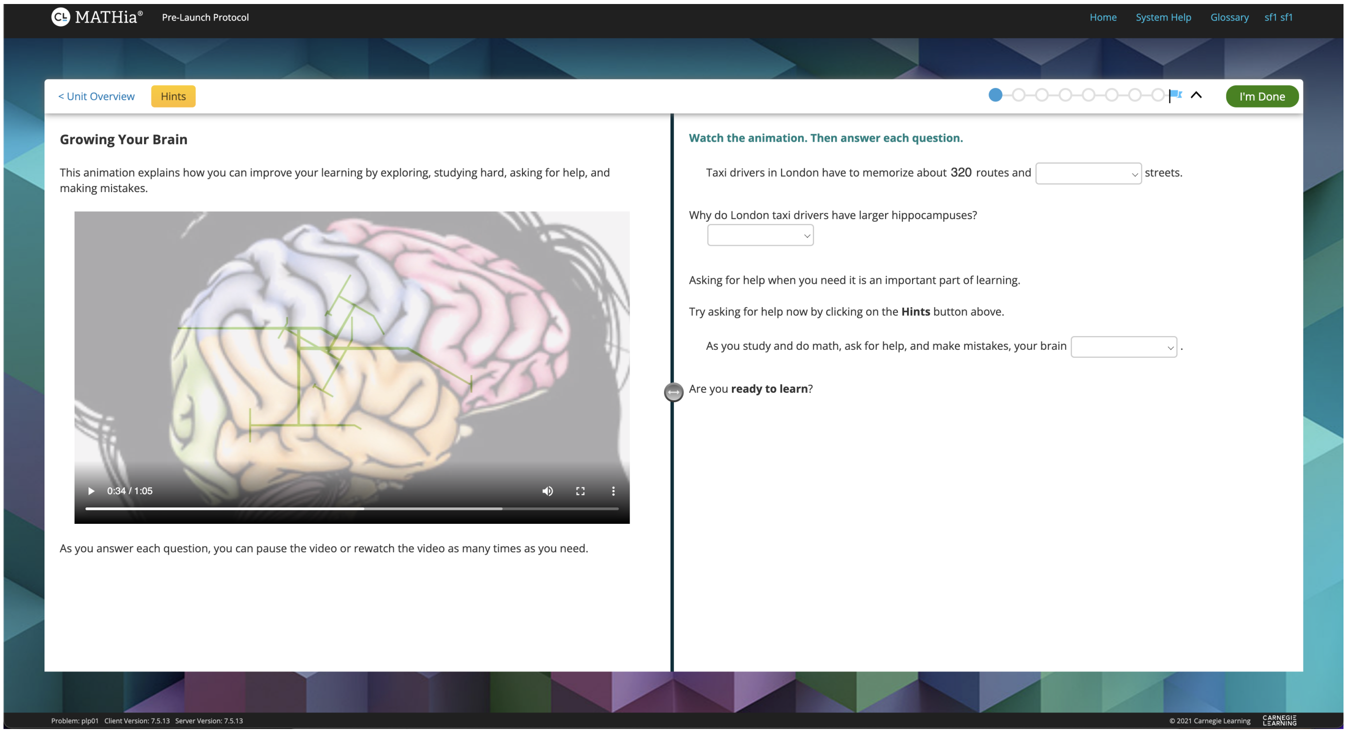
Almoubayyed et al. [1] found that the Pre-Launch Protocol is one of the workspaces in MATHia that are most correlated with end-of-year ELA test scores, compared to their correlations with end-of-year mathematics test scores, across grade levels. Additionally, they were able to build a predictive model of student end-of-year ELA achievement levels by training machine learning models on Pre-Launch Protocol data.

We aim to extend the predictive models of reading ability in MATHia to both explore the generalizability of such a model and to increase trust in it such that it can be used with higher confidence over a large population of users to predict students' reading ability from their interaction with a mathematics ITS.

We use a Multi-Layer Perceptron (MLP) model with identical architecture to the highest-performing model that Almoubayyed et al. developed in [1]. In particular, the model is an MLP with a single hidden layer containing 100 nodes, with a relu activation function and adaptive learning rate. The model is trained with a categorical cross-entropy loss function, optimized by the stochastic gradient-based optimizer defined in [10]. We note that Almoubayyed et al. [1] carried out model exploration with several set-ups. Additionally, while this model explicitly does not provide causal evidence, Almoubayyed et al. do investigate confounding factors in [1]. We do not replicate that work here and we encourage interested readers to refer to [1] for more details on the model details.

We train the model on four sets of features separately: the four sets being the `correct, hint, jit, attempt` defined in Section 3. For each set, there are 36 features corresponding to the steps in the Pre-Launch Protocol. While we retain the model architecture and feature engineering steps, we retrain the model with the following changes:

- We train the model on a binary classification task (passing or failing to pass the state test), rather than achievement levels. This is due to the fact that different states have different numbers of achievement levels, and a binary classifier may be of more practical usefulness. It is possible to use post-processing on a classifier that predicts achievement levels instead of retraining, but we decided that retraining the model on binary classes would be a more consistent implementation across the two districts.
- We treat the Pre-Launch Protocol steps to be the same features regardless of whether a student attempts the Pre-Launch Protocol in a different grade level. The Pre-Launch Protocol itself is identical for any grade level, however, the fact that it is attempted a different grade level may still have predictive signal. We find, however, that the number of students that attempt the Pre-Launch Protocol varies very largely between the



Growing Your Brain

This animation explains how you can improve your learning by exploring, studying hard, asking for help, and making mistakes.

As you answer each question, you can pause the video or rewatch the video as many times as you need.

Watch the animation. Then answer each question.

Taxi drivers in London have to memorize about 320 routes and ______ streets.

Why do London taxi drivers have larger hippocampuses?

Asking for help when you need it is an important part of learning.

Try asking for help now by clicking on the **Hints** button above.

As you study and do math, ask for help, and make mistakes, your brain ______.

Are you **ready to learn**?

Problem: plp01 Client Version: 7.5.13 Server Version: 7.5.13

© 2021 Carnegie Learning

Figure 1: Screenshot of a problem within MATHia's "Pre-Launch Protocol" introductory workspace. The student is presented a brief video animation on the left and then asked questions about the video on the right, serving as an introduction to the MATHia ITS, its user interface, and the adaptive support it can provide.

two districts, and that treating the Pre-Launch Protocol steps to be the same features more appropriate for generalization purposes.

- Almoubayyed et al. [1] developed an ensemble model to combine the four models by taking the mode of the predictions (i.e., a "majority vote" of the four models). Instead, for an ensemble predictive model, we average the predicted probabilities of the four models. Using probabilities allows us to construct a Receiver Operator Characteristic (ROC) curve and avoids situations where the predictions of the four models result in a tie. We refer to this model as `prob`.

We use the ROC curve and the area under the ROC curve (AUC) as metrics to compare models. The ROC curve shows the False Positive Rate (FPR), and the True Positive Rate (TPR), for decision thresholds ranging between 0 and 1 for the classification task. The FPR and TPR are defined as follows:

$$\mathrm{FPR} = \mathrm{FP}/\mathrm{N}$$
$$\mathrm{TPR} = \mathrm{TP}/\mathrm{P}$$

where FP, or False Positives, are defined here as students who are predicted to pass the end-of-year ELA test, but in reality fail to pass it. Conversely, TP, or True Positives, are students who are predicted to pass the end-of-year ELA test, and do indeed pass it. N and P are the total number of negatives and positives respectively in the ground truth dataset.

Analyzing ROC curves allows for choosing specific models with different thresholds depending on the purpose (a lower threshold results in a model with lower FPR and lower TPR, an appropriate choice if minimizing the FPR is a priority. On the other hand, a higher threshold results in a model with higher FPR and higher TPR, an appropriate choice if maximizing the TPR is the priority).

To assess the generalizability of this model, we train and test the models on every combination of training and testing sets. Specifically, we train the 4 (`correct, hint, jit, attempt`) models and compute the ensemble `prob` model on each of the (MW, SE, Combined) training sets, and for each of these models, we test them on each of the (MW, SE, Combined) test sets. This results in 9 combinations (with 4 trained + 1 ensemble model for each of the 9 combinations).

Figure 2 shows the ROC curves for the model trained on the MW dataset and tested on the MW, SE, and Combined datasets, top to bottom respectively. Figure 3 shows the ROC curves for the model trained on the SE dataset and tested on the SE, MW, and Combined datasets, top to bottom respectively. Finally, Figure 4 shows the ROC curves for the model trained on the Combined dataset and tested on the Combined, MW, and SE districts, top to bottom respectively. The ensemble models generally perform significantly better than the four trained models; suggesting that there is a signal gained from combining the four trained models in each case. While a model trained and tested on data from the same school districts performs better, there are no cases where a model tested in a different district performs

Table 1: AUC scores for the ensemble predictors in each case. Each ensemble predictor uses four trained models on each of the MW, SE, and Combined training set, and then each is tested on the MW, SE, and Combined test sets. Models trained and tested on with a dataset from the same district consistently achieve an AUC score of 0.80, while training on one and testing on the other achieves a slightly lower AUC score. Models trained on the Combined training set consistently achieves 0.80 on either test set.

Model	Tested on		
	MW	SE	Combined
MW	0.80	0.76	0.77
SE	0.78	0.80	0.80
Combined	0.80	0.80	0.80

significantly poorer.

Table 1 shows the AUC scores of the ensemble (`prob`) models in each of the 9 cases. We find that the AUC scores range between 0.76 and 0.80. A model trained and tested on the same district, in each of the districts, achieve an AUC of 0.80, while a model trained in one district and tested on the other achieves a slightly lower AUC of 0.76-0.78. Finally, a model trained on both districts achieves an AUC of 0.80 on either district. This suggests that adding data from an additional district makes the model perform better, however, even a model trained in one district and tested in another only slightly underperforms.

Although the district have significantly different performance and base pass rates, the models seem to transfer well without additional changes. Adding data does improve the performance of the models, however, but the performance of these models seems to saturate with an AUC of 0.80 across the two districts.

5. FAIRNESS ASSESSMENTS

Considering how the models perform for different student populations is important to build learners' and other stakeholders' trust in the ITS and ensure that models generalize well over populations of diverse learners nation-wide (and perhaps world-wide). Such considerations are especially important if we are to reach the goal of such embedded assessments playing a role in deployed, real-world ITSs. We look at the ROC curve for each of the ensemble models previously describe (trained on MW, SE, or Combined training sets) when tested on subsets of demographics in each of the test sets. In particular, we look at race and gender information as provided by the school districts. In order to obtain large enough test sets for the demographic subsets, we bifurcate the data into two categories for each demographic. Namely, we look at model performance for white (W) and non-white (NW) students; and for female (F) and male (M) students. We recognize that this bifurcation is broad and does not provide complete information (e.g., on relative model performance for students of different non-white races and for students with different gender identities). We leave more comprehensive and nuanced analyses for important future work.

Figure 5 shows the models' performance when predicting

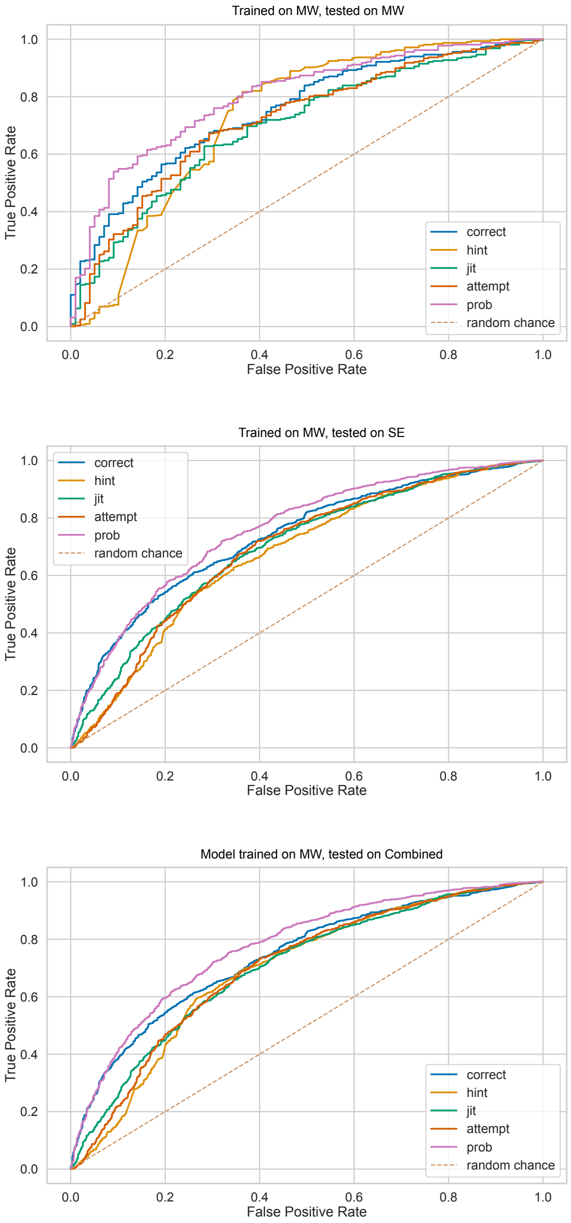
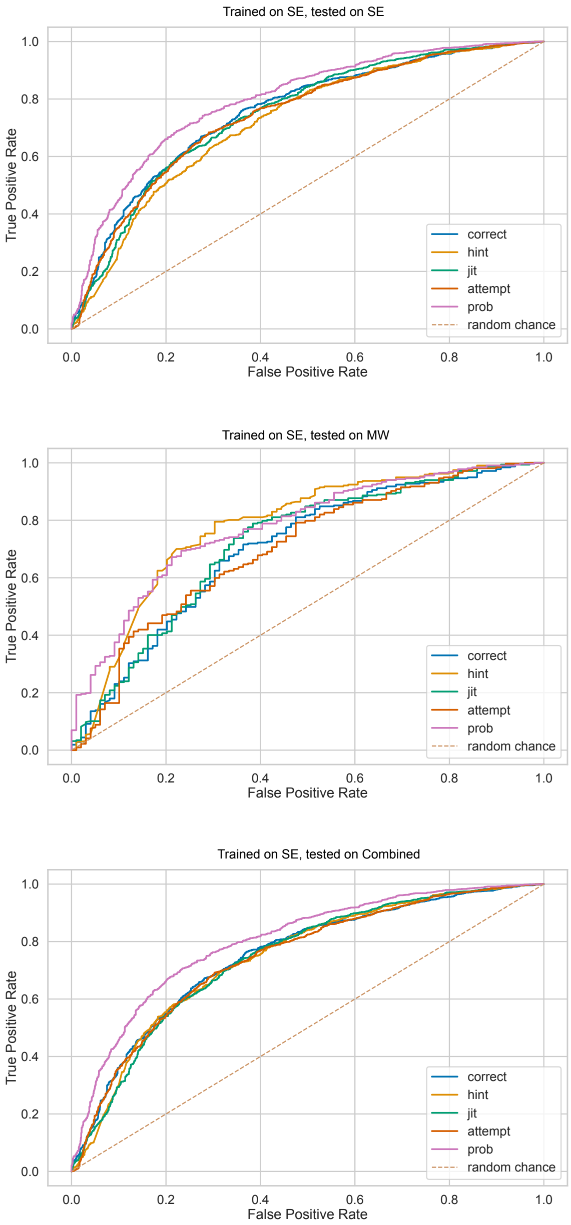


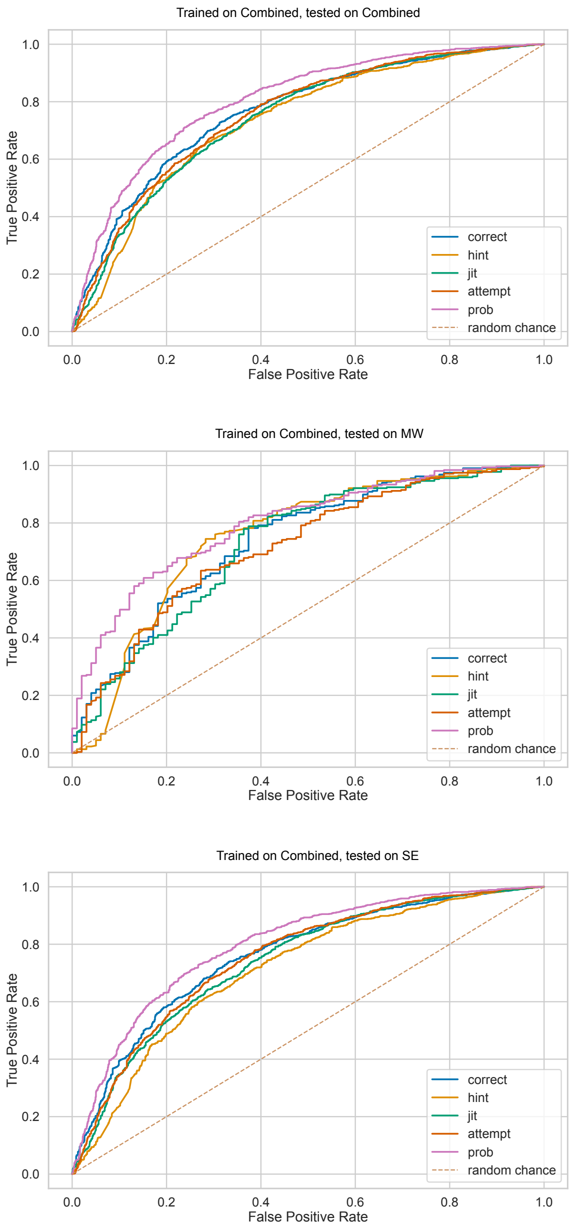
Figure 2: The performance of the four trained models of reading ability and the ensemble model, depicted by the ROC curve of the models, showing the FPR and TPR at different thresholds. These models are trianed on the MW dataset and tested on the MW, SE, and Combined test sets from top to bottom. While the four trained models generally have similar performance, the ensemble model has consistently better performance.


Trained on SE, tested on SE

Trained on SE, tested on MW

Trained on SE, tested on Combined

Figure 3: The performance of the four trained models of reading ability and the ensemble model, depicted by the ROC curve of the models, showing the FPR and TPR at different thresholds. These models are trained on the SE training set and tested on the SE, MW, and Combined test sets from top to bottom. While the four trained models generally have similar performance, the ensemble model has better performance in most cases.


Trained on Combined, tested on Combined

Trained on Combined, tested on MW

Trained on Combined, tested on SE

Figure 4: The performance of the four trained models of reading ability and the ensemble model, depicted by the ROC curve of the models, showing the FPR and TPR at different thresholds. These models are trained on the Combined training set and tested on the Combined, MW, and SE test sets from top to bottom. While the four trained models generally have similar performance, the ensemble model has consistently better performance.

the reading ability of non-white and white students in each of the districts. In all cases, evaluating the model on a test set from the same district yielded similar ROC curve across white and non-white students. However, interestingly, the performance varied significantly when evaluating the models on the other district. Specifically, we see that the model trained on MW data generalized similarly across both non-white and white students, but the model trained on SE and evaluated on MW performs significantly poorer for non-white students compared to white students. Given the relatively similar proportion of white and non-white students in both districts, this suggests that any relatively simple assumption that such similarity ought to lead to similar performance across districts appears flawed. These results are also possibly surprising due to the fact that the MW dataset is significantly smaller in sample size than the SE dataset. In particular, the SE dataset contains over 5 times as many students as the MW dataset.

Similarly for gender, Figure 6 shows the models' performance broken into female and male students. We find a similar trend here, where the model trained on MW generalizes similarly well across female and male students in SE; while there is a significant difference in how the model trained on SE generalizes across female and male students. In particular, we find that the model trained on SE performs significantly poorer when evaluated on male students in MW compared to female students.

Due to the fact that the Combined model is more influenced by data from the (larger) SE district, it performs more similarly to the SE model when broken down by demographics. This leads us to believe that, although the Combined model has a higher AUC on the whole, the MW model might be the better model in practical implementations, due to its similar performance across demographics, at the cost of a slight loss of 0.03-0.04 in AUC performance. Additional data from diverse school districts will be needed to further consider nuances of how models generalize across student populations and the relative fairness characteristics of such generalized models.

While we only consider model performance on different demographics, it may also be valuable to use algorithmic fairness metrics and bias mitigation algorithms. For example, Stinar and Bosch [19] compare the effectiveness of several unfairness mitigation algorithms in the context of mathematics end-of-year state test scores for around 5 million middle school students in Texas; using algorithms such as Disparate Impact Preprocessing [6], Reweighing [4], and Equalized Odds Postprocessing [7].

Disparate Impact Preprocessing, for example, aims to modify the model (by modifying the training data) such that it achieves a Disparate Impact metric closer to unity; where Disparate Impact is defined as

$$DI = \frac{Pr\,(y=1 \mid D=g1)}{Pr\,(y=1 \mid D=g2)}, \tag{1}$$

where y is the target (i.e., $y = 1$ corresponds to passing the state test), and D is the protected class (i.e., the demographic), with $g1$ and $g2$ being two groups in the protected class. When computing the DI metrics on the MW and SE

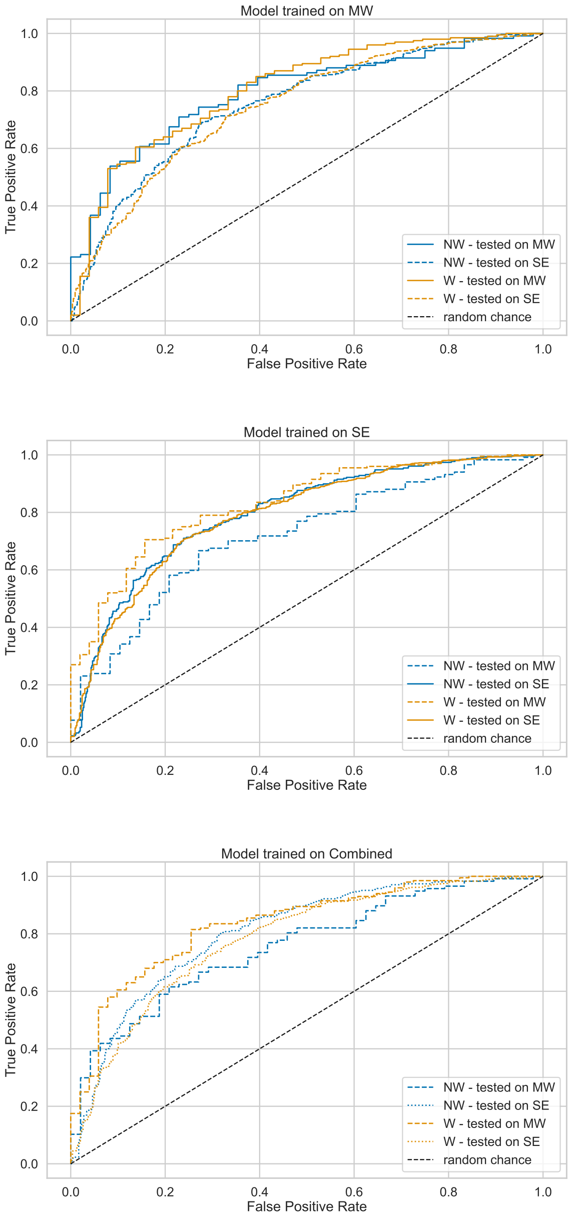


Figure 5: The ROC curve of the ensemble predictors trained on (top to bottom) the MW, SE, and Combined training sets. In the cases where a model was trained on a single district, solid lines correspond to the ROC curve evaluated on a test set that comes from the same district, while dashed lines correspond to evaluation on a test set from the other district. The performance of the predictors are evaluated for white (W) and non-white (NW) students on each of the MW and SE test sets to assess model fairness when generalized to another student population. The plots show that the models trained on the SE and Combined datasets perform significantly poorer when predicting non-white students' reading ability in the MW district. Conversely, the model trained on the MW training set seems to perform similarly well when predicting the reading abilities of both non-white and white students in both districts.

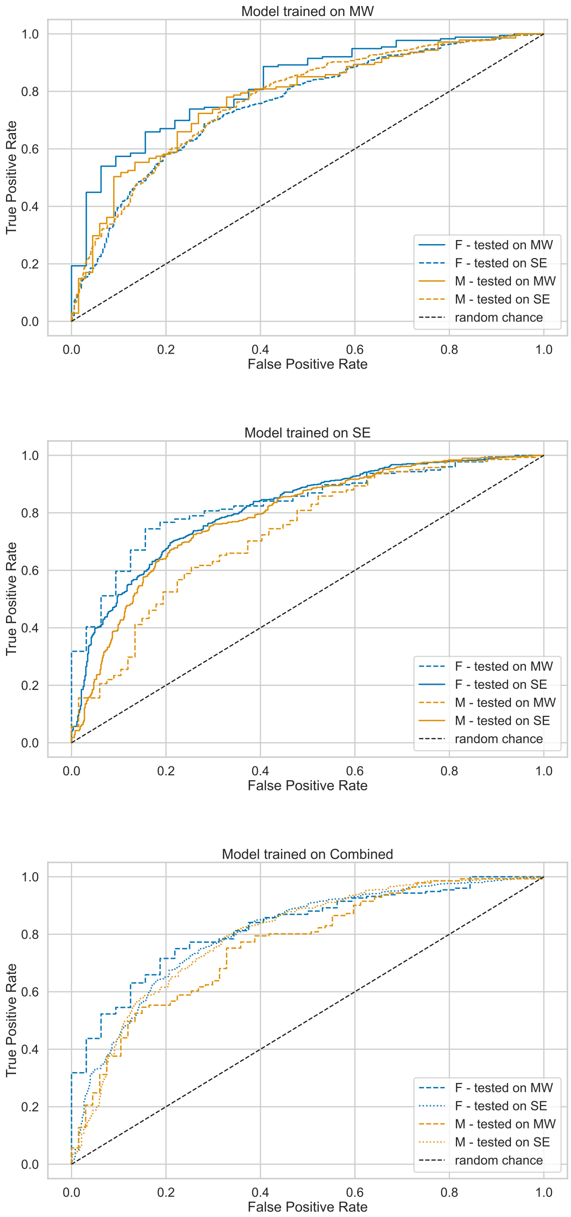


Figure 6: The ROC curve of the ensemble predictors trained on (top to bottom) the MW, SE, and Combined training sets. In the cases where a model was trained on a single district, solid lines correspond to the ROC curve evaluated on a test set that comes from the same district, while dashed lines correspond to evaluation on a test set from the other district. The performance of the predictors are evaluated for female (F) and male (M) students on each of the MW and SE test sets to assess model fairness when generalized to another student population. The plots show that the models trained on the SE and Combined training sets perform significantly poorer when predicting male students' reading ability in the MW district. Conversely, the model trained on the MW training sets seems to perform similarly well when predicting the reading abilities of both female and male students in both districts.

datasets we found that the base rates for the DIs (i.e., the DIs computed on the ground truth data) was in some cases significantly different than 1, and thus there is a trade-off between (a) achieving a DI closer to 1 and (b) achieving better performance on predicting reading ability for students across demographic groups. Upon inspection of the DIs, we do find that the DIs for the model predictions were always slightly closer to unity than the DIs of the test sets. We leave a more comprehensive study of these metrics and whether it is appropriate to use algorithms that aim to alter them to future work.

6. CONCLUSIONS

Results of the present exercise in generalizing a model to predict reading ability built first on data from a school district in the mid-western U.S. [1] to a larger school district in the south-eastern U.S. are promising. We see largely similar predictive performance results (ranging from 0.76 to 0.8 AUC) regardless of whether we learn and/or test models on either of the districts individually or "pool" together or combine data from both districts to create a single dataset for training and testing. These results suggest that such models may be helpful in suggesting relatively "low-stakes" interventions to support readers who may be experiencing difficulty with reading in their mathematics learning in the MATHia software (e.g., behavioral nudges or suggestions to engage with reading supports or possibly directly presenting students with such supports). Additionally, these models are likely to help learning engineers and analysts to better understand whether such supports are working for those they are intended to help (especially if presented across a wide population learners for which data about their reading ability is unavailable).

Our investigations into one facet of algorithmic fairness of the approach we consider leads us to an interesting result: the model trained on a smaller dataset performs better in terms of prediction accuracy across two demographic categories (i.e., a bifurcation of race and gender) we considered while only performing slightly worse overall compared to a model learned over a much larger, pooled dataset. Previous work on data from Cognitive Tutor [21] found a result that was analogous in some ways to the present result, specifically that a model trained over a smaller amount of "high quality" usage data (i.e., students with a lot time using the software and completing content) out-performed models learned over larger populations of students without regard to inclusion criteria for usage. However, the present work considers a much different prediction task, namely ensembled neural network model performance on an end-of-year standardized test outside the target instructional domain of the system, rather than predictions of individual student actions within an ITS. Additionally, the model trained on the larger dataset does out-perform the model trained on a smaller dataset overall; it is just when we begin to consider demographic breakdowns of model performance (as one operationalization of algorithmic fairness, among many) that we start to notice the potential that the model trained on a smaller dataset may be out-performing the model trained on the "larger" dataset. There are other metrics and unfairness mitigation algorithms that have been developed, such as Disparate Impact Preprocessing, Reweighing, and Equalized Odds Postprocessing – we leave a more comprehensive study of these metrics to future work.

While we studied two districts in two states in different regions across the United States, we found that models trained on one or the other have varying performance over different demographics. With data from more states with different demographic make-ups, it would still be interesting to test how these models further generalize, and whether the MW model that generalized well across demographics in both the MW and SE datasets, would also generalize well to districts in more regions.

We believe that taking steps to ensure trust and fairness in predictive models in education are essential when using these models for practical purposes. For example, models that generalize well could be used in A/B testing experiments to predict the reading ability of a large population of students to see how different aspects of personalized learning may work better for them (e.g., by using them to personalize BKT model parameters to students with reading difficulties). An example of such a personalization could, for example, find it more suitable to allow students with predicted reading difficulties to attempt more practice opportunities on mastery content, and vice versa. Such personalization and other adaptive supports may improve student learning and user experience in ITSs, but could also have adverse effects if the predictive ability of the models is unfair towards certain demographics.

We look forward to further engaging with these questions of both generalization and fairness as well as how different goals for prediction are likely to impact appropriate choices for how to operationalize fairness to ensure more trustworthy, equitable, and high-quality learning experiences for all learners.

7. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 2000638.

The research reported here was supported in part by the Institute of Education Sciences, U.S. Department of Education, through grant R324A210289 to Center for Applied Special Technology (CAST). The opinions expressed are those of the authors and do not represent the views of the Institute or the U.S. Department of Education.

Figures in this paper were generated using the open-source Python libraries matplotlib [8] and Seaborn [20]; machine learning models were built and trained using Sci-kit Learn [14]; and statistical computations were done in SciPy [9].

The authors thank April Murphy, Beth Fornauf, Jennifer Reddig, Katie Bevan, Kim Ducharme, Kyle Weldon, Leila Lester, Mark Swartz, Mia Gross, Rae Bastoni, Sarah Galasso, Susan Berman, and Zack Lee at both CAST and Carnegie Learning for helpful discussions and contributions.

8. REFERENCES

[1] H. Almoubayyed, S. E. Fancsali, and S. Ritter. Instruction-embedded assessment for reading ability in adaptive mathematics software. In *Proceedings of the 13th International Conference on Learning Analytics and Knowledge*, LAK '23, New York, NY, USA, 2023. Association for Computing Machinery.

[2] J. R. Anderson and A. T. Corbett. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4:253–278, 1995.

[3] B. S. Bloom. Learning for mastery. instruction and curriculum. volume 1, 1968.

[4] K. Faisal and T. Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33, 2012.

[5] S. E. Fancsali, G. Zheng, Y. Tan, S. Ritter, S. R. Berman, and A. Galyardt. Using embedded formative assessment to predict state summative test scores. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, LAK '18, page 161–170, New York, NY, USA, 2018. Association for Computing Machinery.

[6] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 259–268, New York, NY, USA, 2015. Association for Computing Machinery.

[7] M. Hardt, E. Price, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[8] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[9] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001.

[10] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, page arXiv:1412.6980, Dec. 2014.

[11] K. R. Koedinger, A. T. Corbett, and C. Perfetti. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 36(5):757–798, 2012.

[12] Z. A. Pardos, R. Baker, M. O. S. Pedro, S. M. Gowda, and S. M. Gowda. Affective states and state tests: Investigating how affect and engagement during the school year predict end-of-year learning outcomes. *Journal of learning Analytics*, 1:107–128, 2014.

[13] D. Paunesku, G. M. Walton, C. Romero, E. N. Smith, D. S. Yeager, and C. S. Dweck. Mind-set interventions are a scalable treatment for academic underachievement. *Psychological Science*, 26(6):784–793, 2015. PMID: 25862544.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[15] J. E. Richey, N. G. Lobczowski, P. F. Carvalho, and K. Koedinger. Comprehensive views of math learners: A case for modeling and supporting non-math factors

in adaptive math software. In I. I. Bittencourt, M. Cukurova, K. Muldner, R. Luckin, and E. Millán, editors, *Artificial Intelligence in Education*, pages 460–471, Cham, 2020. Springer International Publishing.

[16] S. Ritter, J. R. Anderson, K. Koedinger, and A. T. Corbett. Cognitive tutor: Applied research in mathematics education. *Psychonomic Bulletin & Review*, 14:249–255, 2007.

[17] S. Ritter, A. Joshi, S. Fancsali, and T. Nixon. Predicting standardized test scores from cognitive tutor interactions. In *EDM*, 2013.

[18] S. Ritter, A. Murphy, S. E. Fancsali, V. Fitkariwala, N. Patel, and J. D. Lomas. Upgrade: An open source tool to support a/b testing in educational software. In *Proceedings of the First Workshop on Educational A/B Testing at Scale*. EdTech Books, 2020.

[19] F. Stinar and N. Bosch. Algorithmic unfairness mitigation in student models: When fairer methods lead to unintended results. In A. Mitrovic and N. Bosch, editors, *Proceedings of the 15th International Conference on Educational Data Mining*, pages 606–611, Durham, United Kingdom, July 2022. International Educational Data Mining Society.

[20] M. Waskom, O. Botvinnik, D. O'Kane, P. Hobson, S. Lukauskas, D. C. Gemperline, T. Augspurger, Y. Halchenko, J. B. Cole, J. Warmenhoven, J. de Ruiter, C. Pye, S. Hoyer, J. Vanderplas, S. Villalba, G. Kunter, E. Quintero, P. Bachant, M. Martin, K. Meyer, A. Miles, Y. Ram, T. Yarkoni, M. L. Williams, C. Evans, C. Fitzgerald, Brian, C. Fonnesbeck, A. Lee, and A. Qalieh. mwaskom/seaborn: v0.8.1, Sept. 2017.

[21] M. V. Yudelson, S. E. Fancsali, S. Ritter, S. R. Berman, T. Nixon, and A. Joshi. Better data beat big data. In *Proceedings of the 7th International Conference on Educational Data Mining*, pages 204–208, 2014.

Partner Keystrokes can Predict Attentional States during Chat-based Conversations

Vishal Kuvar
University of Minnesota
kuvar001@umn.edu

Lauren Flynn
University of Minnesota
flynn598@umn.edu

Laura Allen
University of Minnesota
lallen@umn.edu

Caitlin Mills
University of Minnesota
cmills@umn.edu

ABSTRACT

Computer-mediated social learning contexts have become increasingly popular over the last few years; yet existing models of students' cognitive-affective states have been slower to adopt dyadic interaction data for predictions. Here, we explore the possibility of capitalizing on the inherently social component of collaborative learning by using keystroke log data to make predictions across conversational partners (i.e., using person A's data to make prediction about if person B is mind wandering). Log files from 33 dyads (total N = 66) were used to examine: a) how mind wandering (defined here as task-unrelated thought) during computer-mediated conversations is related to critical outcomes of the conversation (trust, likability, agreement); b) if task-unrelated thought can be predicted by the keystrokes of one's partner; and c) how much data is needed to make predictions by testing various window-sizes of data preceding task-unrelated thought reports. Results indicated a negative relationship between task-unrelated thought and perceptions of the conversation, suggesting that attention is an important factor during computer mediated chat conversations. Finally, in line with our hypothesis, results from mixed effects models showed that one's level of task-unrelated thought was predicted by the keystroke patterns of their conversational partner, but only using small window sizes (5s worth of data).

Keywords

mind wandering, chat, keystrokes, task unrelated thought

1. INTRODUCTION

Imagine you are messaging with a classmate about a homework assignment that is due in your programming class later that day. You exchange rapid messages back and forth, discussing how to debug the problem. You send a last message, but your partner does not immediately reply. Until this moment, your attention had been almost entirely focused on the conversation. But now, in this moment of silence, your attention is captured by thoughts of going to the grocery store once you're finished. You think about how crowded it will probably be, then brainstorm what you want to cook later, and start to think about how you wish you had a sandwich right now. At some point a few minutes later, your friend messages you back and you suddenly realize how far your mind had wandered away from the conversation you two were having.

This example illustrates a critical feature of our attention – namely, that it is constrained by the actions of the people we interact with. In the context of conversation, for example, we are influenced by the content of the messages that our partner sends but also by a variety of more subtle behaviors, such as the timing of the responses themselves. Such timing information is commonly captured via log files in educational technologies, and there is a long history of using this information to predict cognitive and affective states during learning [8]. However, these approaches typically rely on log file information for a particular student to make a prediction about that same student's cognitive state. As our example above illustrates, it may be the case that the behaviors of a conversational partner can provide important information about students' cognitive states that would not otherwise be apparent. With only access to your log data, we would not know why you stopped messaging your partner – was it because you were bored, gave up, or got distracted? Knowing your partner's behaviors helps answer this question perhaps even better than your own.

Here, we expand traditional modeling approaches in the EDM community by examining the predictive power of partner log data to predict attentional states. We designed a computer-mediated conversation task and logged keystroke data from pairs of students while they talked. Periodically, the students were asked to provide self-reports of their attentional states, operationalized here as task-unrelated thought (TUT). Rather than using each student's keystrokes to predict their own attentional state, we test whether they can be predicted from the keystrokes of their partners. Assessing the feasibility of using partner data to predict cognitive states is particularly timely today where interactions amongst individuals are increasingly occurring online and may continue in this direction with the advent of large language model based chatbots (e.g., Chat-GPT). It is therefore important that we consider new methodologies that rely on

V. Kuvar, L. Flynn, L. Allen, and C. Mills. Partner keystrokes can predict attentional states during chat-based conversations. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 217–223, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115697

numerous sources of log data beyond those of the individual student, which can provide opportunities to model and respond to student attention.

1.1 Related Work

1.1.1 Task-Unrelated Thought

TUT tends to occur around 20-30% during computerized reading [9], 30-40% during online lectures [22], and 20% while interacting with an intelligent tutoring system [16]. Importantly, TUT frequency has consistently demonstrated a negative relationship with affective valence [18] and learning outcomes [9, 25]. Given the frequently negative consequences of TUT on learning, researchers and educational technology developers have placed a strong emphasis on the development of models that can detect when a student has gone off task based on log data that can be readily integrated within a system. These models have relied on a variety of different sources of data to date, such as reading times, eye gaze, and EEG signals [11, 10]. These detectors can then be used to increase adaptivity and personalization in educational technologies. For example, recent work has shown that a TUT-sensitive intervention was effective for promoting long-term comprehension compared to a control group who did not receive the interventions at the moment they needed them [17].

Despite the substantial body of work on TUT during learning, it has rarely been examined in collaborative contexts, such as computer-mediated communication (CMC) or interactive learning contexts where chats are the most common form of communication among students (or between teachers/bots and students). One exception is recent work demonstrating that TUT occurs quite frequently when students are chatting with one another on computers in separate locations; instances of TUT were also correlated negative affective valence and other variables during the chat, providing initial evidence that it might be an important indicator of chat outcomes [4]. However, this study currently exists in isolation, leaving large gaps in our understanding of if and how TUT matters during conversations.

Given the nascent work in this area, our first research questions center around if and how such instances of TUT relate to perceived conversational outcomes; that is, what is the benefit of knowing whether students are off-task, and is it predictive of outcomes we care about in collaborative learning? Answering these questions will provide a baseline for future work in the context of EDM – namely motivating why we should consider modeling attention in the context of student computer mediated chats. A few variables that are of particular interest along these lines are likability and trust [19, 23]. However, trust is often difficult to measure directly or in real-time, so proxy stealth measurements that are linked to trust could provide "early warnings" for interventions. Indeed, for any chat-based system to be effective, these variables will be critically important to understand and detect early on so that students don't disengage before it's too late.

At the same time, if TUT is predictive of key outcomes, then we also need to understand effective ways to model it as chat conversations unfold. In our context, we are focused on understanding which behaviors, that can be readily extracted in chat data, may be used to predict cognitive states – particularly ones that capture the inherent social interdependence that exists within dyadic chats. This may be quite timely to explore given that CMC – especially remote, real-time chats – is becoming increasingly used in educational settings.

1.1.2 Keystroke Data

We chose to use keystroke data to explore this question given past work showing that keystroke log files are able to provide fine-grained temporal information about students' language production. For instance, the number of keys a student presses at the beginning of a writing task can provide insights into the degree to which their ideas were developed before they began the task. Similarly, a high number of backspaces may indicate that the student was revising their ideas in the moment. Keystroke features such as these have been linked to numerous factors related to learning, such as emotions [1, 3], reflective evaluation [24], and the quality of written product itself [14, 1, 5].

Predictive models using keystroke data have predominantly focused on writing tasks completed by single students, such as argumentative essays (see [6] for a review). However, there is some work in the CMC literature that examines the role of message timing in conversational success; for example, research indicates that shorter pauses with fewer keystrokes are associated with increased trust in your conversational partner [13]. These prior studies provide a foundation for work using keystrokes in CMC contexts; however, many questions remain unanswered. Relevant to the current work, how might the keystrokes of our partners relate to our own attentional states? As illustrated above, it is likely that the rhythm of our conversational partner may have a direct influence on our own attentional states; however, research is still needed to provide a more formal account of how partners' behaviors relate to cognitive and affective states.

1.2 Overview and Novelty of Current Work

There is no shortage of educational technologies that can detect and respond to an individual's cognitive states. Still, relatively few studies have leveraged the inherent interdependence between individuals in social contexts to inform such technologies. As collaborative learning becomes increasingly popular, understanding these links may open new doors to predictive modeling in collaborative tasks. Towards this goal, we expand the traditional application of log files to make cross-partner predictions of attention in a dyadic conversation from readily available keystroke log files from 33 dyads. In doing so, we answer the following research questions: a) how TUT during computer-mediated conversations is related to critical outcomes of the conversation (trust, likability, agreement); b) if TUT can be predicted by the keystrokes of one's partner; and c) how much data is needed to make predictions by testing various window-sizes of data preceding task-unrelated thought reports.

2. METHODS

2.1 Data Collection

2.1.1 Participants

We collected data from participants using an online platform called Prolific, where participants were paid to engage

Table 1: Keystroke features and descriptions.

Feature Type	Keystroke Feature	Description	Mean (Std.)	
			5s	15s
Non-message	Verbosity	Number of keystrokes in the window	20.34 (6.229)	47.98 (16.63)
	Backspace Frequency	Number of times the backspace key was hit in the window	0.002 (0.020)	0.005 (0.028)
	Maximum Latency	Maximum difference between two successive keystrokes in the window	5713 (4578)	15257 (10933)
	Median Latency	Median difference between two successive keystrokes in window	1152 (2616)	864.8 (2485)
Message	Inter-Word Time	Mean time between consecutive words in the recreated message	334.6 (120.9)	619.4 (272.3)
	Inter-Sentence Time	Mean time between consecutive sentence in the recreated message	354.3 (1194)	588.9 (1736)
	Number of Words	Number of words in recreated message (separated by space key)	2.563 (0.903)	5.997 (2.244)
	Maximum Sentence Length(# of Keystrokes)	Maximum number of words logged to type a sentence	18.79 (1.665)	41.33 (14.53)

in our chat-based study. All participants (n=218) locations were limited to the United States and the United Kingdom. Prolific has been shown to be a reliable source for data collection and can yield more diverse datasets in terms of participant background and age. Participants had a mean age of 34.016 (SD=11.45). 73.7% were female, 24.7% male, and 1.6% reported being non-binary. 79.1% participants were Caucasian, 11% Asian, 3.4% Hispanic, 1.1% Black, 0.5% American Indian and 4.9% as 'other.'

2.1.2 Chat Platform

We built our own online chat platform where two participants would be automatically matched and converse with each other while answering in-situ "thought probes" about whether they were experiencing TUT throughout the chat session. The chat was designed to be much like an large online discussion, where two students may be randomly paired with each other and asked to chat. The entire conversation lasted a total of 16 minutes. During the conversation, all keystrokes and their associated timestamps were logged. We attempted to keep the conversations relatively open-ended to mimic real-life chats between students. Each conversational pair was given one of three different instructions for the conversation to create diversity in the chats (i.e. so any findings could not be attributed to forcing a single type of chat/topic): 1) high constraints, where the participants were asked explicitly to learn about remedies for the common cold from one another; 2) low constraint, in which participants were asked to discuss medically relevant topics; 3) no constraints, where participants were simply asked to chat with each other. Note that this manipulation was not necessarily intended to lead to differences in our dependent variables (and we find no significant differences in the key variables across conditions); rather, we include it to test whether our findings generalize across multiple topics. However, for the sake of caution, we included topical condition as a "control" variable in all of the analyses presented.

After the task, participants were redirected to a survey page where they answered questions about their demographics, valence, and arousal. Valence measures how positive to negative participants feel, whereas arousal captures participant's level of activation, or how sleepy to active they feel. The survey also consisted of questions about trust, likability, and agreement the participant felt towards their partner.

2.1.3 Thought Probes

Six brief thought probes were administered pseudo-randomly throughout the duration of the conversation. Both participants saw the probes simultaneously about every two and a half minutes. The probe read: "On a scale of 7, please select a number that most reflects your attention on the current task right now. 1 being you are completely focused on task and 7 being you are not focused on the task at all." This thought probe method is the gold-standard in TUT research, particularly in educational contexts [9]. Although there are inevitably some limitations that come with using self-reports, this method has been validated numerous times in different contexts (lab settings, online research, classrooms). Results suggest that thought probes do not have a negative influence on task performance, and the responses have reliable correlates [21]. In our study, both conversational partners received the probe simultaneously. Message sending was disabled until the participant responded to the probe.

2.1.4 Trust, Likability, and Agreement Questionnaires

Participants answered questions about trust, likability, and agreement immediately after talking to their chat partner. An 11 item scale designed by McCallister [15] was used to measure trust. Likability was measured using a modified version of the Rysen Likability Scale (RLS; [20]). Out of the original 11 items on that scale, we chose five to include that were relevant to online interactions. Items on both questionnaires were reported using a 7-point Likert scale. Five questions were constructed to measure the agreement of chat perceptions between participants. An example of a question was "I was interested in my partner's viewpoint." Participants reported on a 9-point Likert Scale. Participants answers were added for each scale and these sums were used in subsequent analyses.

2.2 Data Processing and Analyses

2.2.1 Data Cleaning

We collected 218 keystroke files. Due to a glitch in the server, 33 of the files logged keystrokes as "Unidentified." Additionally, four files contained fewer than 200 keystrokes. These 37 files could not be used for the feature extraction process and were dropped. Given that our second research question was based on interdependence, it was imperative that were able to align data from both conversational partners. However, given that we were unable to align for these same 37 participants, their respective conversational partners was also dropped. We also removed all pairs of participants who did not get a total of six probes due to an error in the triggering system (N = 39 pairs). The final total number of participants that could be used for analysis was 66 (33 pairs).

2.2.2 Window Creation

Our primary goal is to assess whether keystroke patterns can be used to predict the attention of someone's conversational partner. We thus needed to align keystroke patterns with thought reports in a time-sensitive manner. That is, we needed to extract keystroke data from a "window" leading up to the thought report (but not including the report itself). This windowed approach is commonly used for detecting TUT in real-time [11], but this is the first time it has been applied in a dyadic context, where we take data from one person to make a prediction about the other.

We created windows leading up to each thought probe using two window sizes that have been successful in prior research: 5s and 15s [11]. We created these windows in the time leading up to the probe, such that keystroke data extracted from the chat would predict their partner's future level of TUT. The window was defined by the nearest keystroke to the thought probe. That is, if a person did not type in the 5s window immediately preceding the thought probe, the algorithm would instead search for the closest keystroke and begin the window at this point. This approach was necessary given the dyadic nature of the task, where conversational partners often take consecutive turns. If the keystroke preceding the probe was typed outside of the window size, only that keystroke would be included in the window. This results in the features extracted from these windows to have low values, compared to when keystrokes are present.

2.2.3 Features

Once a window was defined, keystroke features were extracted and classified into two categories: message and non-message features (see Table 1). Message features require the recreation of the message within the window, whereas non-message features use the raw keystrokes. The non-message features that we selected were based on Bixler and D'Mello [3]. Messages in the window were recreated by processing the keystrokes in the window in a sequential manner. A space key indicated a new word. A period, exclamation mark, or question mark indicated the end of a sentence. If a backspace key was encountered in a window, the previously typed key would be deleted.

Table 2: Correlation matrix of self-reported TUT, arousal, and partner perception.

	Prop. TUT reports	Valence	Arousal	Trust	Likability
Valence	-.201				
Arousal	.117	.194			
Trust	-.372*	.083	.075		
Likability	-.261·	.311·	.122	.600**	
Agreement	-.312·	.193	.009	.742**	.521**

**p<0.01, *p<0.05, .p<0.1

2.2.4 Analytical Approach

The lme4 package [2] in R was used to create mixed-effects models. We extracted features from the raw keystrokes and used the standardized version of them as data. Models were fitted with random intercepts, with the participant acting as the random effect. This accounted for within-subjects variance in the responses. For this analysis, the independent variables were the individual keystroke features of a participant. The dependent variable was the response of their partner for the TUT probe. We report the unstandardized regression coefficient (B), p-value, 95% confidence intervals, and standardized β.

3. RESULTS

Given that TUT has not been studied often in the context of CMC, a major contribution of this work is evidence that participants' average level of TUT was 2.40. This indicates they were predominantly on task relative to the midpoint of the scale (3.5 on a 1 to 7 scale), but nevertheless went off task quite a bit (SD = 1.36). Participants also seemed to feel generally positive with an average affective valence of 3.53 (SD = .78), and they appear to moderately trust and agree with their conversational partners.

3.1 Does TUT relate to the perceptions of conversation?

TUT has a consistent negative relationship for affective valence and learning outcomes, but these are almost explicitly studied in individual tasks. Our first research question was thus to explore how levels of TUT relate to affective states as well as perceptions of the chat. These correlations help address a basic question: is TUT worth detecting in the context of conversations? Table 2 presents the Pearson correlation values between variables, where each person's own level of TUT was correlated with their self-reported affect and perceptions of the chat. Out of the 66 participants, only 62 were used to calculate these values. Data for the remaining four were missing. First, we replicated the typically observed negative relationship between TUT and affective valence positive [18]. Second, we also observed significant correlations between TUT and perceptions of the chat – namely, increased TUT was associated with less trust, likability, and agreement with your conversational partner.

3.2 Do keystrokes predict partner TUT and at what window size?

Table 3: Results of mixed effects models.

Predictors (Keystroke features)	Attention level of conversational partner									
	5s window					15s window				
	B	β	p	95% CI		B	β	p	95% CI	
				Lower	Upper				Lower	Upper
Intercept	3.02	0.00	<0.001	2.61	3.43	3.02	0.00	<0.001	2.61	3.43
Verbosity	0.17	0.08	0.04	0.00	0.35	0.08	0.04	0.42	-0.11	0.26
Backspace frequency	0.00	0.00	0.99	-0.16	0.16	-0.01	-0.01	0.81	-0.18	0.14
Maximum latency	0.08	0.04	0.31	-0.08	0.24	-0.02	-0.02	0.67	-0.20	0.13
Median latency	0.13	0.06	0.11	-0.03	0.29	0.07	0.07	0.05	-0.01	0.31
Inter-word time	0.10	0.05	0.20	-0.06	0.26	-0.03	-0.03	0.46	-0.23	0.10
Inter-sentence time	-0.17	-0.08	0.03	-0.33	-0.01	-0.06	-0.06	0.08	-0.30	0.02
Word count	0.15	0.07	0.08	-0.02	0.32	0.03	0.03	0.50	-0.12	0.24
Maximum sentence length (Keystrokes)	0.18	0.08	0.04	0.01	0.36	0.03	0.03	0.42	-0.11	0.25

Table 3 provides the full results for all regression models. For the 5s window, three keystroke features significantly predicted partner's level of TUT: verbosity, inter-sentence time and maximum sentence length. Verbosity was positively related to partner TUT, suggesting that when someone types for longer periods (with more keystrokes), their partner's mind is more likely to drift off-task. A similar relationship was observed between maximum sentence length and TUT. Taken together, these relationships indicate that when individuals produce longer messages, their partners may be more likely to go off-task, perhaps while waiting for their partner to complete their thought.

Unlike the keystroke features above, the inter-sentence time feature was negatively related to partner TUT, with a one standard deviation increase in inter-sentence length corresponding to a 0.17 decrease in TUT. The inter-sentence time feature provides information about when partners pause between the sentences they produce. Thus, it provides some context for the pauses in the chat rather than simply examining all pauses regardless of when they occur. The negative relationship between this feature and TUT suggests that certain types of pauses may be more or less important for a partner's TUT. In particular, if an individual pauses after drafting a full sentence, it is more likely the case that their partner now has a complete idea that they can reflect upon and respond to, rather than something more incomplete. This is a particularly compelling interpretation, given that overall pause times (latencies) were unrelated to partner TUT reports.

Importantly, all of the significant relationships that we observed were for keystroke features calculated at the 5s window, not at the 15s window. This suggests that the keystroke features were predictive of partner TUT rates, but only for those recorded immediately before the probe was delivered. We cannot make causal claims about why this is the case; however, a strong possibility is that the window sizes for keystroke features are sensitive to the specific type of conversation that is taking place. Here, students were asked to have a conversation while not engaging in any other tasks – this single-task paradigm resulted in relatively rapid turn-taking between the partners.

Finally, it is worth noting that even the significant models revealed a relatively small effect in terms of the variance explained by the fixed effects effects in our linear mixed effects regressions. The predictors verbosity, inter-sentence time, and maximum sentence length had a conditional R^2 value of 0.007, 0.006, and 0.007, respectively – leaving an opportunity to refine such models in future work.

4. DISCUSSION

Collaborative learning environments are inherently social. One person's actions will inevitably influence others. The current study leveraged this interdependence among individuals in a conversational setting in order to determine if log file data can be helpful for predicting cross-partner cognitive states. Our main hypothesis was that, in a conversational setting, one partner's keystrokes may be indicative of their partner's attentional state. Taken together, our results support this hypothesis – highlighting the idea that incorporating the interdependence between individuals into predictive models may be beneficial for adaptive educational technologies supporting collaborative learning. Specifically, we demonstrate that keystrokes are one such feature that can be leveraged to make these predictions in the context of computer-mediated conversations.

Verbosity, inter-sentence time, and maximum sentence length were the three keystroke features that were reliably predictive of partner TUT. However, it is worth noting that these features were only significant within relatively short (5s) window sizes. The window sizes at which keystroke features should be calculated are likely to depend on the context of the conversation taking place; thus, when examining keystroke data, researchers should extract features at multiple window sizes to determine which are most appropriate for their context. A second implication of our study for future research using keystroke data is the use of non-message and message features. We found that inter-sentence time was a reliable predictor of partner TUT ratings, but there were no relations to overall pause time. This indicates that keystroke features may benefit from the addition of contextual information from the conversation itself. For example, pauses after highly emotional messages may reflect different processes than those after relatively mundane messages, such as making plans or asking simple questions.

Our study adds to the growing body of work suggesting that keystrokes are an indicator of cognitive states. Keystroke features, such as the ones extracted here, are readily available in most log files, yet are not commonly used in multimodal models. It may therefore be worthwhile adding this feature to increase predictive power in both individual and collaborative settings. Our paper is focused on the latter context; as such, we believe there may be particular promise in interactive group contexts as individual models may not always reveal the whole story: is someone interacting less because they are bored, frustrated, or confused? One's own data may not be the best way to answer this question; perhaps the person cannot get a word in because the conversation is moving too fast, or perhaps everyone else has stopped responding. Adding keystrokes to predictive models can thus allow for the social component to be included in a more explicit way, facilitating time-sensitive nudges or subtle feedback to conversational partners on their interactions.

A final set of findings emerged to suggest that TUT is worth monitoring in CMC. People seemed to experience TUT quite often during computer-mediated conversations, in line with previous work showing it is ubiquitous in almost all aspects of our lives, including educational activities [12, 25]. Not only does it happen often, these experiences do not appear to be particularly positive; increased levels of TUT were consistently and negatively related to trust, likability, and agreement amongst partners in ours study. This complements prior work that links TUT to negative affect and clinical conditions – underscoring a potential need to detect it and respond in educational technologies.

An interesting possibility to consider, particularly as chatbots (e.g., Chat-GPT) are likely to continue rapidly evolving, is how our findings can be expanded in a chatbot setting. With chatbots becoming more knowledgeable and accessible, a possibility that bots can be used in education cannot be ignored. Future work may consider exploring how chatbots mimicking keystroke patterns that are associated with lower levels of TUT may influence engagement, and thus learning outcomes [25]. There is already some evidence that predictive models of TUT (using one's own keystrokes) are accurate during dyadic CMC interactions, and that the results generalize to chatbot settings; expanding this work in more ecological and with multiple conversational participants' keystrokes would likely be fruitful (i.e. even beyond dyadic interactions to group interactions).

Like most research, ours is not without limitations. For example, we created our own chat platform and did not provide participants with an explicit learning goal. Although we took care to vary the topic, replicating our results under different goal conditions will be an important next step. We were also somewhat limited with sample size, limiting our scalability. Nevertheless, our analytical approach provides proof-of-concept for the usefulness of using partner data. Future research may also wish to improve our models by including content-dependent features, such as the conversational topic. These limitations and caveats notwithstanding, we believe that "attending to attention" [7] will be helpful in building technologies that can facilitate effective online collaboration.

5. REFERENCES

[1] L. K. Allen, M. E. Jacovina, M. Dascalu, R. D. Roscoe, K. M. Kent, A. D. Likens, and D. S. McNamara. {ENTER}ing the Time Series {SPACE}: Uncovering the Writing Process through Keystroke Analyses. Technical report, International Educational Data Mining Society, 2016. Publication Title: International Educational Data Mining Society ERIC Number: ED592674.

[2] D. Bates, M. Mächler, B. Bolker, and S. Walker. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67:1–48, Oct. 2015.

[3] R. Bixler and S. D'Mello. Detecting boredom and engagement during writing with keystroke analysis, task appraisals, and stable traits. In *Proceedings of the 2013 international conference on Intelligent user interfaces - IUI '13*, page 225, Santa Monica, California, USA, 2013. ACM Press.

[4] A. Colby, A. Wong, L. Allen, A. Kun, and C. Mills. Perceived group identity alters task-unrelated thought and attentional divergence during conversations. *Cognitive Science*, 47(1):e13236:1–23, 2023.

[5] R. Conijn, C. Cook, M. van Zaanen, and L. Van Waes. Early prediction of writing quality using keystroke logging. *International Journal of Artificial Intelligence in Education*, 32(4):835–866, Dec. 2022.

[6] P. Deane and M. Zhang. Automated Writing Process Analysis. In D. Yan, A. A. Rupp, and P. W. Foltz, editors, *Handbook of Automated Scoring*, pages 347–364. Chapman and Hall/CRC, 1 edition, Feb. 2020.

[7] S. D'Mello, K. Kopp, R. E. Bixler, and N. Bosch. Attending to Attention: Detecting and Combating Mind Wandering during Computerized Reading. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '16*, pages 1661–1669, San Jose, California, USA, 2016. ACM Press.

[8] S. K. D'mello and J. Kory. A review and meta-analysis of multimodal affect detection systems. *ACM Comput. Surv.*, 47(3):1–36, feb 2015.

[9] S. K. D'Mello and C. S. Mills. Mind wandering during reading: An interdisciplinary and integrative review of psychological, computing, and intervention research and theory. *Language and Linguistics Compass*, 15(4):e12412:1–32, 2021.

[10] H. W. Dong, C. Mills, R. T. Knight, and J. W. Y. Kam. Detection of mind wandering using EEG: Within and across individuals. *PLOS ONE*, 16(5):1–18, May 2021. Publisher: Public Library of Science.

[11] M. Faber, R. Bixler, and S. K. D'Mello. An automated behavioral measure of mind wandering during computerized reading. *Behav Res*, 50(1):134–150, Feb. 2018.

[12] S. Hutt, K. Krasich, C. Mills, N. Bosch, S. White, J. R. Brockmole, and S. K. D'Mello. Automated gaze-based mind wandering detection during computerized learning in classrooms. *User Model User-Adap Inter*, 29(4):821–867, Sept. 2019.

[13] Y. M. Kalman, L. E. Scissors, A. J. Gill, and D. Gergle. Online chronemics convey social information. *Computers in Human Behavior*, 29(3):1260–1269, May 2013.

[14] A. D. Likens, A. Likens, L. K. Allen, and D. S. McNamara. Keystroke Dynamics Predict Essay Quality. In *Annual Meeting of the Cognitive Science Society*, page 6, July 2017.

[15] D. J. McAllister. Affect- and cognition-based trust as foundations for interpersonal cooperation in organizations. *The Academy of Management Journal*, 38(1):24–59, 1995.

[16] C. Mills, S. D'Mello, N. Bosch, and A. M. Olney. Mind Wandering During Learning with an Intelligent Tutoring System. In C. Conati, N. Heffernan, A. Mitrovic, and M. F. Verdejo, editors, *Artificial Intelligence in Education*, volume 9112, pages 267–276, Cham, 2015. Springer International Publishing.

[17] C. Mills, J. Gregg, R. Bixler, and S. K. D'Mello. Eye-Mind reader: an intelligent reading interface that promotes long-term comprehension by detecting and responding to mind wandering. *Human–Computer Interaction*, 36(4):306–332, July 2021.

[18] C. Mills, A. R. Porter, J. R. Andrews-Hanna, K. Christoff, and A. Colby. How task-unrelated and freely moving thought relate to affect: Evidence for dissociable patterns in everyday life. *Emotion (Washington, D.C.)*, 21(5):1029–1040, Aug. 2021.

[19] E. Molleman, A. Nauta, and B. P. Buunk. Social Comparison-Based Thoughts in Groups: Their Associations With Interpersonal Trust and Learning Outcomes. *Journal of Applied Social Psychology*, 37(6):1163–1180, 2007.

[20] S. Reysen. CONSTRUCTION OF A NEW SCALE: THE REYSEN LIKABILITY SCALE. *Social Behavior and Personality: an international journal*, 33(2):201–208, Jan. 2005.

[21] A.-L. Schubert, G. T. Frischkorn, and J. Rummel. The validity of the online thought-probing procedure of mind wandering is not threatened by variations of probe rate and probe framing. *Psychological Research*, 84(7):1846–1856, Oct. 2020.

[22] K. K. Szpunar, N. Y. Khan, and D. L. Schacter. Interpolated memory tests reduce mind wandering and improve learning of online lectures. *Proceedings of the National Academy of Sciences*, 110(16):6313–6317, Apr. 2013.

[23] S. W. Uranowitz and K. O. Doyle. Being liked and teaching: The effects and bases of personal likability in college instruction. *Res High Educ*, 9(1):15–41, Mar. 1978.

[24] A. Wengelin, M. Torrance, K. Holmqvist, S. Simpson, D. Galbraith, V. Johansson, and R. Johansson. Combined eyetracking and keystroke-logging methods for studying cognitive processes in text production. *Behavior Research Methods*, 41(2):337–351, May 2009.

[25] A. Y. Wong, S. L. Smith, C. A. McGrath, L. E. Flynn, and C. Mills. Task-unrelated thought during educational activities: A meta-analysis of its occurrence and relationship with learning. *Contemporary Educational Psychology*, 71:102098:1–18, Oct. 2022.

Meta-Learning for Better Learning: Using Meta-Learning Methods to Automatically Label Exam Questions with Detailed Learning Objectives

Amir Zur[*]
Stanford University
Department of Computer Science
amirzur@stanford.edu

Isaac Applebaum[*]
Stanford University
Department of Biology
iapple23@stanford.edu

Jocelyn Elizabeth Nardo
Stanford University
Graduate School of Education
jnardo@stanford.edu

Dory DeWeese
Stanford University
Department of Chemistry
dory@stanford.edu

Sameer Sundrani
Stanford University
Department of Biomedical Computation
sundrani@stanford.edu

Shima Salehi
Stanford University
Graduate School of Education
salehi@stanford.edu

ABSTRACT

Detailed learning objectives foster an effective and equitable learning environment by clarifying what instructors expect students to learn, rather than requiring students to use prior knowledge to infer these expectations. When questions are labeled with relevant learning goals, students understand which skills are tested by those questions. Labeling also helps instructors provide personalized feedback based on the learning objectives each student struggles to master. However, developing detailed learning objectives is time-consuming, making many instructors unable to pursue it. Labeling course questions with learning objectives can be even more time-intensive. To address this challenge, we develop a benchmark for automatically labeling questions with learning objectives. The benchmark comprises 4,875 questions and 1,267 expert-verified learning objectives from college physics and chemistry textbooks. This dataset provides a large library of learning objectives, and, to the best of our knowledge, is the first benchmark to measure performance on labeling questions with learning objectives. We use meta-learning methods to train classifiers and test them against our benchmark in a few-shot classification setting. These classifiers achieve acceptable performance on a test set with previously unseen questions (AUC 0.84), as well as a course with previously unseen questions and unseen learning objectives (AUC 0.84). Our work facilitates labeling questions with learning objectives to help instructors provide better feedback and create equitable learning environments[1].

A. Zur, I. Applebaum, J. Nardo, D. DeWeese, S. Sundrani, and S. Salehi. Meta-learning for better learning: Using meta-learning methods to automatically label exam questions with detailed learning objectives. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 224–233, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115677

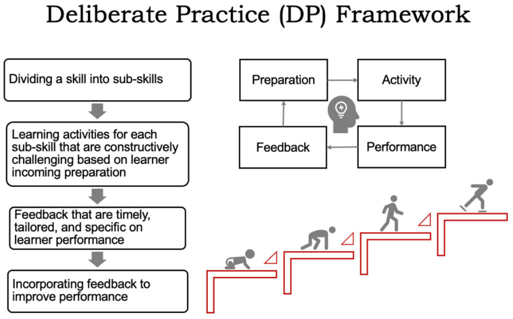

Deliberate Practice (DP) Framework
Dividing a skill into sub-skills
Learning activities for each sub-skill that are constructively challenging based on learner incoming preparation
Feedback that are timely, tailored, and specific on learner performance
Incorporating feedback to improve performance
Preparation
Activity
Feedback
Performance

Figure 1: Deliberate practice framework adapted from [6].

Keywords

educational equity, assessment, learning objectives, pedagogical tool, personalized feedback, meta-learning

1. INTRODUCTION

Ericsson and colleagues argue that instructors can maximize their students' learning and improvement over time by facilitating deliberate practice [6]. To facilitate deliberate practice, instructors should break targeted skills into separate subskills, and design learning activities to practice each subskill in a way that takes students' prior knowledge into account. Importantly, students should receive "immediate informative feedback" about their performance on these tasks. Afterwards, students should be given the opportunity to improve their performance, whether by revising their work or by applying what they learned to a similar task. Our version of the deliberate practice framework is shown in Figure 1 [16].As shown by Glaser and Chi, breaking down larger skills into smaller subskills can also facilitate development of mental schema to organize domain knowledge, a key characteristic of expertise [4]. Deliberate practice provides a useful theoretical framework for understanding the benefits of detailed learning objectives and labeling course materials with these objectives.

[*] Equal contribution.

[1] Repository: `https://github.com/AmirZur/smartstem-ai`

Implementing deliberate practice in a classroom environment requires providing effective feedback. Ramaprasad argues that true feedback entails clearly articulating a goal, providing information about the gap between current performance and this goal, and ensuring that this information is used to bring current performance closer to the goal [19]. Ruiz-Primo and colleagues apply these criteria in their study of formative assessment [20]. Ruiz-Primo et al. argue that instructors should address three questions when teaching: "Where are we going?", "Where are we now?", and "How will we get there?". Completing the "Where are we going?" step involves writing learning objectives and clarifying what is considered evidence of achieving these learning objectives. Detailed learning objectives therefore provide a clear goal to measure student performance against. The "Where are we now?" step involves assessment, which provides a measure of students' current and prior knowledge. If assessments are intentionally designed around relevant learning objectives, and questions are labeled with the learning objectives they assess, this clarifies the gap between students' performance and the goals defined by the learning objectives. The "How will we get there?" step involves instructors tailoring their instructional practices to meet students' specific needs, which can include reinforcing concepts that a student may be struggling with and allowing students to revise their work [20, 24]. Labeling questions with learning objectives allows instructors to analyze the specific areas where each student needs help, and more effectively tailor instruction to the needs of their students. Finally, exam questions labeled with detailed learning objectives can particularly benefit students with less prior preparation, since these students may be less able to independently identify the skills tested by questions [17, 21, 22].

However, developing detailed learning objectives is difficult and time-consuming, which causes many educators to avoid writing learning objectives altogether, or to write only a few general learning objectives that do not communicate the specific skills that they expect students to demonstrate. Labeling questions with the relevant learning objectives is even more challenging and time-intensive, making it harder to provide effective feedback to students. To address such challenges, this work uses data mining and AI techniques to help instructors reap the benefits of learning objectives to facilitate equitable learning outcomes. We develop a benchmark for automatically labeling questions with learning objectives, using a custom dataset comprising a total of 4,875 questions and 1,267 expert-verified learning objectives drawn from four OpenStax college physics and chemistry textbooks, a widely-used college chemistry textbook, and Stanford University's general chemistry course materials (hereafter, Chem 31A). This dataset provides educators with a large library of learning objectives and questions, and, to the best of our knowledge, is the first benchmark to measure performance on labeling questions with detailed learning objectives. We use our benchmark to train and test three different types of classifiers: a multi-class multi-label (MCML) classifier, a ProtoTransformer, and a classifier adapted from GPT-3 embeddings. The ProtoTransformers and GPT-3 classifiers perform few-shot classification, a meta-learning task in which a classifier predicts the class of an input out of previously unseen classes given a few example items for each class (see Section 2.1 for more detail). Our results show that these few-shot classifiers achieve acceptable performance on our held-out test set, which consists of previously unseen course questions (AUC 0.84). Furthermore, the ProtoTransformer and GPT-3 classifiers generalize to a held-out course, which consists of previously unseen course questions and previously unseen learning objectives (AUC 0.84). Our work facilitates labeling questions with learning objectives, which can help instructors to incorporate learning objectives into their courses, provide better feedback, and create more equitable learning environments for students.

2. RELATED WORKS

Although previous research has supported educators' efforts to generate and analyze learning objectives [3, 12, 18], there has been limited research on facilitating the automatic labeling of questions with learning objectives. Some relevant work has been done on automatic exam grading, which can be viewed as labeling questions with rubric items [14, 30, 31]. Our work follows most closely the ProtoTransformer [31], which uses prototypical networks [25] to train a transformer-based model [29] to automatically grade computer science exams. In this section, we reintroduce the problem of few-shot classification, expand on the ProtoTransformer approach to this problem, and compare it with two other classification methods: multiclass-multilabel (MCML) classifiers and GPT-3 adapted as a few-shot classifier [2].

2.1 Few-Shot Classification

Few-shot classification is a meta-learning task in which, given a few training examples of each class, a classifier must adapt to predict new classes that were not previously seen in training [10, 11, 15, 25]. In our work, we formulate the task of labeling questions with learning objectives as a few-shot classification problem in which the classifier is trained to label questions with learning objectives, and the set of learning objectives and questions can vary from course to course.

In our learning setting, we consider a distribution D consisting of task indicators, input examples, and output labels. Formally, let $(t, x, y) \sim D$ be a task indicator, input example, and label drawn from a distribution of meta-learning tasks. We consider learning objectives $t \in T$ to be task indicators, and questions $x \in X$ to be model inputs. The task label, $y \in Y = \{0, 1\}$, is such that $y = 1$ if question x is labeled with learning objective t, and $y = 0$ otherwise. In this work, our goal is to train a model f_θ to accurately predict y given question x and learning objective t.

To perform few-shot classification, we are given a support set of k examples for each of the n prediction classes, $S = \{(x_1, y_1), \dots, (x_{k\times n}, y_{k\times n})\}$. This work considers binary classification ($n = 2$), and so we interpret our support set as follows: S contains k examples of questions that are labeled with learning objective t (i.e., examples where $y = 1$), and k examples of questions *not* labeled with learning objective t (i.e., examples where $y = 0$). The goal of a few-shot classifier f_θ is, given S and an unlabeled question x, to classify whether or not it should be labeled with learning objective t. Note that the classes predicted by a few-shot classifier may not be the same between training time and inference time. In fact, the classes differ with each task type. That is, for the same input question x, the correct label may sometimes

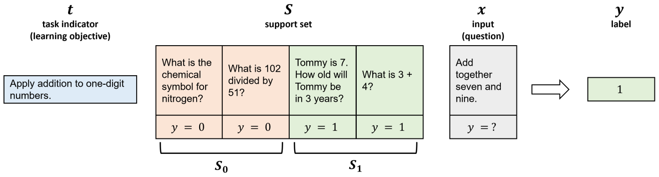

t
task indicator
(learning objective)
Apply addition to one-digit numbers.
S
support set
What is the chemical symbol for nitrogen?
$y = 0$
What is 102 divided by 51?
$y = 0$
Tommy is 7. How old will Tommy be in 3 years?
$y = 1$
What is 3 + 4?
$y = 1$
S_0
S_1
x
input
(question)
Add together seven and nine.
$y = ?$
y
label
1

Figure 2: Example few-shot learning tasks in our setting, with $k = 2$. Each task consists of a task indicator (learning objective) t, a support set S containing k negative examples of questions (i.e., questions *not* labeled with learning objective t) and k positive examples of questions (i.e., questions labeled with learning objective t), an input question x, and a label y, where $y = 1$ if x should be labeled with t, and 0 otherwise.

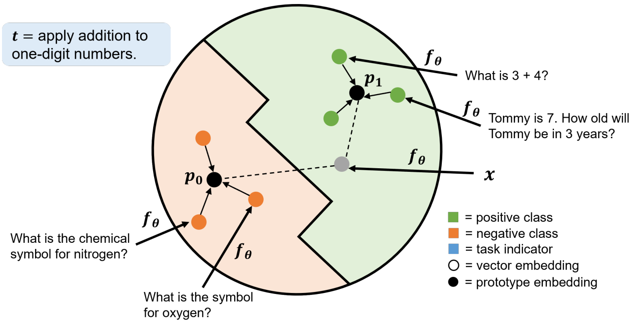


Figure 3: Visualization of prototypical networks adapted from Snell et al., 2017 [25], with $n = 2, k = 3$. For each class (i.e., questions labeled with t and questions not labeled with t) we are provided three examples of questions. The network f_θ maps each question to an embedding, and computes the prototype p_c of each class by averaging the class embeddings. A new input question, x, is then classified by taking the closest prototype to its embedding (in this figure, x is labeled 1).

be 0 and sometimes be 1, depending on the learning objective indicator t. Hence, a few-shot classifier must rely on the support set S, which consists of example questions for each class (i.e., examples of questions that should and shouldn't be labeled with t). As long as we can provide a support set S, a few-shot classifier can classify new questions with new learning objectives that do not appear during training. An example of few-shot classification is provided in Figure 2.

2.2 Prototypical Networks

One method for few-shot learning classification is prototypical networks [25], which serves as the basis of the ProtoTransformer and adapted GPT-3 classifiers [2, 31]. Prototypical networks embed inputs into vectors, such that similar inputs are closer together within the network's embedding space. For each prediction class, prototypical networks create a prototype embedding by taking the average embedding of all support examples in that class. New inputs are then classified by finding the closest class prototype within the network's embedding space.

Here we formalize the prototypical network algorithm. Given a support set S and class label c, let $S_c = \{(x_i, y_i) \in S \mid y_i = c\}$ be all examples of class c in S. For example, S_0 contains all questions in the support set *not* labeled with learning objective t. The prototype embedding of class c is $p_c = \frac{1}{k}\sum_{x_i \in S_c} f_\theta(x_i)$. That is, the prototype of each class represents the mean embedding of inputs with the same class label. The prototypical network then predicts the label y of an unseen question x by taking a softmax over the distance of the model's embedding of x, $f_\theta(x)$, from each prototype p_c (see Equation 1). In our setting, this is equivalent to asking, "Is the network's embedding of our question closer to the average embedding of questions labeled with learning objective t or questions not labeled with t?"

$$p(y = c \mid x) = \frac{\exp(-\text{dist}(f_\theta(x), p_c))}{\sum_{c'} \exp(-\text{dist}(f_\theta(x), p_{c'}))} \tag{1}$$

The network is trained to minimize the negative log-probability $-\log p(y = y \mid x)$ of the true class y. In our setting, dist is the L$_2$ distance function.

2.3 ProtoTransformer Classifier

The ProtoTransformer classifier is a prototypical network with a transformers-based architecture [31]. One key feature of the ProtoTransformer classifier is its ability to incorporate textual information from the task indicator (i.e., learning objective), which we expand upon in this section.

Prototypical networks generalize to previously unseen input examples (i.e., course questions) and to previously unseen task indicators (i.e., learning objectives). However, representing learning objectives as task indicators does not allow our model to utilize textual information from the learning objectives themselves. Note that as illustrated in Figure 3, prototypical networks do not make use of the content of the task indicator t – they only use the positive and negative examples of the task – in order to classify a new input x. The ProtoTransformer classifier addresses this problem by incorporating information from the task indicator in its embedding layer. The ProtoTransformer uses a separate embedding function g_ϕ, a pre-trained transformers model [29] with frozen parameters, to compute a vector representation of the learning objective, and adds this vector representation to the beginning of its model embeddings. The resulting embedded representation (i.e., learning objective token concatenated with question embedding) is passed into the transformers architecture, so that the interaction between the learning objective and question information can be used to construct an output vector. That is, the ProtoTransformer treats a learning objective as a sort of "task token," which informs the model of the relation between the input question and its learning objective. An example ProtoTransformer embedding layer is illustrated in Figure 4.

2.4 MCML Classifier

Another approach to labeling questions with learning objectives utilizes multi-class multi-label (MCML) classifiers [28]. MCML classifiers, given an input question x, learn to predict a binary vector y with an entry for each learning objective t, such that the t-th entry of y is 1 for all learning objectives that x should be labeled with, and 0 otherwise. Although MCML classifiers are not few-shot learners, in that they do not use the support set S in their predictions, they contribute to a field of prior research on fine-tuning transformer classifiers [29]. In our setting, we fine-tune an MCML model with a transformers-based architecture on a collected

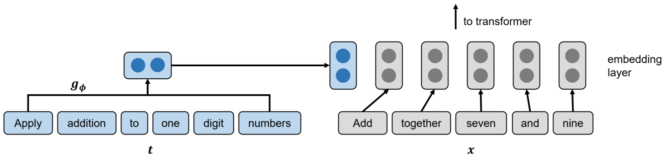


Figure 4: Figure adapted from Wu et al., 2021 [31], illustrating the embedding-space architecture used in our model in order to incorporate textual information from the task indicator (learning objective) t.

Table 1: Cross-comparison of classifiers used in this work to label questions with learning objectives. Both ProtoTransformers and GPT-3 perform few-shot classification, while MCML does not. Although GPT-3 does not require any fine-tuning or additional training, it is not freely accessible and must be accessed through a monetized API.

Classifier	Few-Shot	Fine-tuned	Free Access
ProtoTransformer	✓	✓	✓
MCML	✗	✓	✓
GPT-3	✓	✗	✗

dataset of questions labeled with learning objectives. The MCML model serves as our baseline, since it is not trained by the meta-learning algorithm for prototypical networks.

2.5 GPT-3 Classifier

Recent research has investigated the potential of large language models to perform few-shot classification without additional training [1, 2]. In our work, we adapt a recent large generative model, GPT-3 [2], as a prototypical network. That is, when run on an input question x, the adapted GPT-3 model output $f_{\text{GPT-3}}(x)$ is the activation of the last hidden layer within the GPT-3 model. The final layer activation constitutes a vector representation that is used to compute the prototype embedding for each class within the support set during few-shot classification. One advantage of GPT-3 is that since it is a large pre-trained language model, we expect its hidden layers to provide rich embeddings of text across various domains, including our collected course questions and learning objectives. Hence, GPT-3 does not require any fine-tuning nor additional training in our setting. On the other hand, GPT-3 is not publicly available, and, as of time of writing, is only accessible through a monetized API. This restriction does not apply to the ProtoTransformer and MCML classifiers, and is further discussed in Section 7.2.

In summary, we are not aware of prior research which has focused on the task of labeling course questions with learning objectives. Nevertheless, recent research on ProtoTransformer, MCML classification, and large language models such as GPT-3 provides avenues for developing models to label questions with learning objectives from previously unseen courses. We summarize the key attributes of prototypical networks, MCML classifiers, and few-shot GPT-3 as pertains to our work in Table 1.

3. METHODOLOGY

Our work introduces a benchmark for automatically labeling questions with learning objectives, on which we analyze the ProtoTransformer, MCML, and adapted GPT-3 classifiers. In this section, we provide details on the benchmark data collection process and the classifier training process.

3.1 Benchmark Creation

We collected 4,875 questions and 1,267 expert-verified learning objectives from four publicly available OpenStax textbooks (Chemistry 2e [8], University Physics I, II, and III [13]), a commonly-used university chemistry textbook (Principles of Chemistry 3rd edition [27]), and a Stanford University introductory chemistry course (Chem 31A). The questions from all OpenStax textbooks, as well as from the university chemistry textbook, are labeled with the corresponding list of learning objectives included in each textbook. To collect data from Chem 31A, we worked with members of the course teaching team to manually develop a list of 75 specific learning objectives for the course. For reliability, we independently labeled 30 exam questions (30 percent of the total dataset) and reached an agreement of 98 percent with a Cronbach's alpha score of 0.90, consistent with excellent inter-rater reliability [26]. We then labeled 98 exam questions from the 2021 offering of the course, consisting of four assessments, with the relevant learning objectives from our list. After coding all 98 exam questions, we found that only 53 of our learning objectives were covered by these exam questions. Although other repositories of learning objectives are available [3, 12, 18], to the best of our knowledge this is the first dataset to allow for training and benchmarking machine learning models on the labeling of course questions with relevant learning objectives. Example data points from our dataset can be found in Table 4, and are further discussed in Section A in the Appendix.

3.2 Classifier Training

Our main contribution in this work, besides the creation of the benchmark, is a collection of classifiers (ProtoTransformer, MCML, GPT-3), trained and tested on our benchmark for labeling questions with learning objectives. We use a ProtoTransformer with a BERT architecture, keeping the default settings from the original paper [29] (∼110M parameters). We train the ProtoTransformer with an Adam optimizer [9] and a learning rate of 1×10^{-5} for 8 epochs on our training dataset, which consists of ∼950 of k-shot classification tasks. The k value during training is 5, although we vary k during inference time. Our implementation of MCML is a BERT model (same hyperparameters as the ProtoTransformer) fine-tuned on our training data. We train the MCML classifier with an Adam optimizer and a learning rate of 1×10^{-5} for 5 epochs on our training dataset. Lastly, we adapt GPT-3 using the OpenAI curie model [2] (∼6.7B parameters) as described in Section 2.5, without additional training.

4. EXPERIMENTS

4.1 Experiment 1: Held-Out Test Set

We evaluate our model on a held-out test set, which consists of previously unseen learning objectives. Although questions were shared with the training set, the support set and query set consist of previously unseen combinations of questions and corresponding learning objectives, hence constituting a previously unseen task. In our benchmark test dataset, positive examples (i.e., question-learning objective pairs in which the question is labeled by that learning objective) are balanced with negative examples (i.e., question-learning objective pairs in which the question is not labeled by that learning objective).

4.2 Experiment 2: Held-Out Course

Our second experiment considers using the trained classifiers to automatically label questions with learning objectives on a full course. We use a held-out course, Chem 31A, which consists of 53 previously unseen learning objectives and 98 previously unseen questions. We note that the MCML classifier is inapplicable in this setting, since the learning objective class labels are unavailable to it during training. Hence, we only compare the ProtoTransformer and GPT-3 classifiers. Unlike the test set, our held-out course is unbalanced with regards to positive and negative examples. A course question in Chem 31A is labeled with one to eight learning objectives of the total 53 available; therefore, our held-out course data is skewed towards negative examples.

Due to the imbalance in our dataset and multiple learning objective labels per question, we evaluate models with respect to ROC-AUC and F1 scores in addition to accuracy [7, 23]. The ROC-AUC metric considers a moving decision boundary, allowing us to better interpret the tradeoff between precision, or the ability to predict a short list of learning objectives that match the true learning objectives per question (with the risk of excluding true learning objectives), and recall, or the ability to predict all learning true objectives per question (with the risk of providing a long list containing unrelated learning objectives). Likewise, the F1 score balances precision and recall in its computation, accounting for class imbalances. We use the accuracy, AUC, and F1 evaluation metrics in both the held-out test and held-out course experiments.

4.3 Experiment 3: Recall Over Top-m

Due to the imbalanced nature of our held-out course dataset, in which questions are labeled with one to three of 53 learning objectives, we expect our model to over-predict the list of learning objectives with which to label a question (i.e., generate an overly-long list of candidate learning objectives for a single question). Interestingly, error types in our setting are imbalanced as well. A false positive (type I error) in our setting occurs when our model labels a question with an incorrect learning objective, meaning that an educator would need to filter a longer list of predicted learning objectives in order to label a question. Meanwhile, a false negative (type II error) occurs when our model fails to label a question with one of its correct learning objectives, meaning that an educator would need to search through the entire course list of learning objectives in order to find the correct learning objective. As a result, false negative errors would be far more time-consuming for an educator to correct. Hence, our last experiment considers recall, which measures a classifier's protection against false negative errors. We consider a graph of recall over top-m, where m represents the number of positive labels that the classifier assigns (i.e., the number of learning objectives labeled per question), chosen by taking the m learning objectives with the highest probability predicted by the classifier. A higher m represents more post-processing on behalf of educators (e.g., filtering from a list of five vs. ten predicted learning objectives); meanwhile, a higher recall score indicates that a greater percentage of true learning objectives are contained in the list of m learning objectives.

Table 2: Comparison of classifier performances on held-out test set. Highest scores are in bold.

k	Classifier	Accuracy	AUC	F1
0	MCML	0.52 ± .02	0.51 ± .00	0.34 ± .00
1	GPT-3	0.53 ± .02	0.55 ± .03	0.43 ± .02
	ProtoTransformer	0.68 ± .01	0.79 ± .02	0.63 ± .02
2	GPT-3	0.53 ± .02	0.54 ± .03	0.43 ± .02
	ProtoTransformer	0.74 ± .01	0.83 ± .02	0.71 ± .02
5	GPT-3	0.52 ± .02	0.53 ± .03	0.44 ± .02
	ProtoTransformer	**0.77 ± .01**	**0.84 ± .01**	**0.74 ± .01**

5. RESULTS

Below we detail classifier performance across each of our experiments. We also provide example classifier outputs in Table 5, and a preliminary qualitative analysis of classifier behavior in Section B in the Appendix.

5.1 Experiment 1: Held-Out Test Set

We report accuracy, ROC-AUC, and F1 scores on our held-out test set for the ProtoTransformer, MCML, and GPT-3 classifiers, across varying values of k (see Table 2). Higher values of k denote more examples provided to a few-shot learner per classification (in our case, more examples of questions that are labeled with a certain learning objective), and hence a greater manual effort to label questions with learning objectives. Since the MCML model is not a few-shot classifier, we treat it as a zero-shot classifier with $k = 0$. Both the ProtoTransformer and GPT-3 classifiers significantly outperform the MCML classifier, with the ProtoTransformer achieving the strongest performance at $k = 5$ (AUC of 0.84).

5.2 Experiment 2: Held-Out Course

We compare the ProtoTransformer and GPT-3 classifiers on a held-out course, Chem 31A, which consists of previously unseen questions and previously unseen learning objectives. We report results across varying values of k, corresponding to the number of example questions per learning objective that the classifier requires in order to label the remaining course's questions (see Table 3). The ProtoTransformer model requires at least $k = 1$ example per learning objective. Meanwhile, GPT-3 can be used as a zero-shot learning model, where each learning objective class is represented by the GPT-3 embedding of the learning objective itself. In this experiment, GPT-3 outperforms the ProtoTransformer classifier, achieving an AUC of 0.80 on the $k = 1$ setting.

5.3 Experiment 3: Recall Over Top-m

Figure 5 illustrates the trade-off between m, the total number of learning objective labels that a model assigns to a single input question, and the model's recall. A larger m means that an educator would need to filter between a longer list of outputted learning objectives. Meanwhile, a larger recall means that the list of outputted learning objectives contains

Table 3: Comparison of classifier performances on held-out course. Highest scores are in bold.

k	Classifier	Accuracy	AUC	F1
0	GPT-3	0.76 ± .02	0.66 ± .05	0.49 ± .02
1	GPT-3	0.63 ± .03	**0.80 ± .04**	0.46 ± .02
	ProtoTransformer	0.47 ± .05	0.73 ± .04	0.36 ± .03
2	GPT-3	0.77 ± .03	0.75 ± .05	0.55 ± .03
	ProtoTransformer	0.63 ± .02	0.74 ± .05	0.46 ± .02
5	GPT-3	**0.84 ± .03**	0.79 ± .05	**0.61 ± .03**
	ProtoTransformer	0.66 ± .03	0.77 ± .04	0.48 ± .02

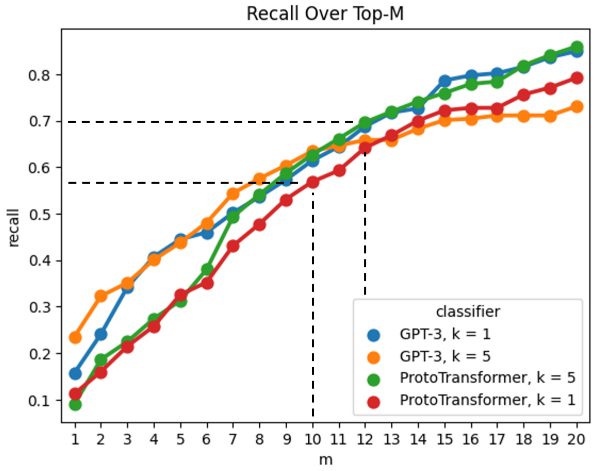


Figure 5: Model performance, measured as recall of true learning objectives, over m, or the number of learning objectives predicted by the model.

a higher percent of the learning objectives that match the input question. The plot below shows that at larger m values and $k = 5$, the ProtoTransformer model achieves stronger recall than GPT-3. Nevertheless, GPT-3 achieves higher recall at $k = 1$ and $k = 5$ when limited to lower m values (between 8 and 12).

6. DISCUSSION

The main contribution of our work is a custom benchmark and a collection of classifiers trained on our benchmark to facilitate the process of labeling questions with learning objectives. Our classifiers generalize to a held-out course, Chem 31A, with previously unseen questions and learning objectives. We therefore believe that these classifiers can be applied to other courses to help educators introduce learning objectives in their classrooms.

Our experiments evaluate an MCML classifier and two few-shot classifiers, the adapted GPT-3 and ProtoTransformer. When benchmarked on our held-out test set, which consists of previously unseen course questions and seen learning objectives, the ProtoTransformer significantly outperforms both the GPT-3 and MCML classifiers (AUC of 0.77, 0.52, 0.52, respectively). These results suggest that the ProtoTransformer model generalizes to new few-shot classification tasks, and is suitable for use in courses that share similar learning objectives to our dataset (e.g. university-level STEM courses). Meanwhile, the MCML classifier, without the ability to perform few-shot classification, is not as suitable as the meta-learning approaches of the adapted GPT-3 and ProtoTransformer classifiers.

In the second experiment, we analyze classifier performance on our held-out course, Chem 31A, with previously unseen questions and learning objectives. The results of this experiment demonstrate how few-shot classifiers could be used to automatically label new questions with new learning objectives. Both the ProtoTransformer and GPT-3 classifiers achieve acceptable performance on the $k = 5$ setting (AUC 0.77, 0.79, respectively), in which the instructor would need to provide 5 examples of questions for each learning objective. Interestingly, the GPT-3 classifier tested on the $k = 1$ setting – requiring only one example question per learning objective – achieves the strongest AUC score of 0.80. This is a promising result which showcases the capability of GPT-3 to perform few-shot classification without any additional training. Therefore, the GPT-3 classifiers can be a better choice for labeling questions with learning objectives of a new course with unseen learning objectives and questions.

Our recall over top-m plot, seen in Figure 5, confirms the strength of GPT-3 as a few-shot classifier. The ProtoTransformer achieves the strongest recall given a larger m value, meaning that when allowed to tag a question with 20 learning objectives, the ProtoTransformer is the most likely model to include the correct learning objectives within the list of 20 predictions. Nevertheless, the $k = 5$ GPT-3 classifier achieves acceptable recall (0.63) at $m = 10$, striking a balance between overly-long lists of learning objectives and the retrieval of accurate learning objectives. We note that the GPT-3 classifier in the $k = 1$ setting, which requires less manual question labeling on behalf of an educator, achieves an acceptable recall (0.69) at $m = 12$. Figure 5, then, illustrates the power of the ProtoTransformer and adapted GPT-3 classifiers to label previously unseen questions with previously unseen learning objectives.

7. LIMITATIONS

7.1 Benchmark Limitations

While the results in this work suggest that our dataset of questions labeled with relevant, specific learning objectives is a reliable and useful benchmark, it is limited by the specificity of the OpenStax learning objectives and their corresponding questions. An inspection of the OpenStax portion of our benchmark, which constitutes the training dataset for our models, reveals that a question is labeled by each of its subchapter's learning objectives, not all of which may be relevant. This limitation also means that questions spanning multiple concepts are only labeled with learning objectives from a particular course unit. See Section A in the Appendix for a detailed analysis of the OpenStax dataset.

The fact that OpenStax questions are not labeled with subsidiary learning objectives from other sub-chapters while Chem 31A questions are labeled with such subsidiary learning objectives may help explain why classifiers trained on OpenStax questions perform better on the held-out course, Chem 31A, than on the held-out OpenStax dataset (see Tables 2 and 3). Another potential explanation is that the OpenStax dataset contains 1,267 learning objectives while the Chem 31A dataset contains 53 learning objectives, meaning that the classifiers need to choose from fewer learning objectives when labeling Chem 31A questions. The smaller number of learning goals in Chem 31A is likely more representative of a single course, rather than a textbook that could be used to teach a series of courses. Therefore, the OpenStax dataset could pose a more challenging labeling task than the intended use case of assisting course instructors.

Another notable limitation is that we automatically collected the OpenStax data using a custom web scraping program (available on our GitHub repository), without any data preprocessing such as removing special unicode characters or addressing typos. While this limitation does not seem to prevent our classifers from performing effectively on the OpenStax dataset, systematically correcting typos could improve classifier performance and increase the dataset's usefulness to both instructors and researchers.

7.2 Classifier Limitations

Key differences between our ProtoTransformer and GPT-3 models, beyond classification performance, include model size and accessibility. Our trained ProtoTransformer model is an order of magnitude smaller than the respective GPT-3 classifier (∼110M vs. ∼6.7B parameters), and is freely accessible for usage and further training. As of time of writing, GPT-3 is only accessible through a monetized API[2], and, partly due to its size, is not readily available for additional training. Hence, we encourage further use and exploration of the ProtoTransformer classifier.

At the same time, we acknowledge that although achieving an AUC score of 77% on a held-out course is promising, the ProtoTransformer classifier may not be accurate enough for use in all introductory STEM courses. We hope that future research using our benchmark will improve classifier accuracy, and potentially generalizability to different course subjects. For immediate use in classroom settings, we recommend that instructors investigate model outputs carefully, and filter its predicted learning objectives down to the ones most relevant to the question at hand. Instructors can use Figure 5 to determine the number of learning objectives that they would like to filter from (we recommend an m between 12 and 16). Furthermore, future research could ensemble multiple classifiers together (e.g. ProtoTransformer at $k = 5$, GPT-3 at $k = 1$, and GPT-3 at $k = 5$) in order to improve classifier accuracy [5].

Lastly, our preliminary qualitative analysis of example model behavior (see Section B in the Appendix) suggests that the performance of our few-shot classifiers is limited by the provided input. That is, access to high-quality support examples during inference time (i.e., questions that are already labeled with learning objectives by the instructor) is essential for accurate prediction. Future work on decreasing k while maintaining high accuracy, along with work on identifying learning objectives that do not receive as much course coverage, can significantly enhance the capabilities of our classifiers.

8. FUTURE WORKS

Our results enable many exciting future works for educators in chemistry, physics, and other STEM fields. By facilitating the process of labeling questions with learning objectives, we aim to help educators introduce learning objectives into their classrooms and label course materials with these objectives, actions that support students towards mastery-based learning approaches and promote equity [17]. Since our classifiers can label new questions with existing learning objectives and our dataset includes expert-verified learning objectives from multiple fields, our classifiers can be used to generate lists of detailed learning objectives for courses that currently have none. Rather than designing learning objectives from scratch, instructors could use our classifiers to label their existing course materials with the relevant learning objectives from our dataset. The list of learning objectives chosen by the classifiers can serve as a draft list of learning objectives for the course, which instructors can adapt to fit their needs. To facilitate these applications, our research team is currently developing an interactive web-based tool to allow instructors to experiment with our trained classifiers. This tool will allow instructors to automatically label their own questions with learning objectives from our datasets, or with other learning objectives that they provide. In addition, the tool will allow instructors to choose the value of hyperparameters such as m, the number of learning objectives that they would like the model to recommend as potentially relevant to each question, in order to best align with their needs. Furthermore, the performance of GPT-3 in this work as a prototypical neural network and as a zero-shot classifier motivates further exploration of GPT-3 as a meta-learning model, and its use within educational domains. Lastly, we encourage data scientists and educators to use and expand on our dataset of learning objectives, which we believe is the first benchmark of its kind to label questions with learning objectives.

9. CONCLUSIONS

Questions labeled with learning objectives can help students use feedback to better navigate their course, particularly benefiting students with less prior preparation. However, the task of labeling questions with learning objectives is time-consuming, making many instructors unable to pursue it. In this paper, we introduce a benchmark and trained classifiers for automatically labeling course questions with learning objectives. We show that meta-learning classifiers trained on our benchmark achieve acceptable performance on a test set with previously unseen questions (AUC 0.84), as well as a previously unseen course (AUC 0.84). We believe that our work, and future research in this realm, can support educators by facilitating the process of developing and utilizing learning objectives in their courses to create more effective and equitable learning environments.

10. REFERENCES

[1] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are

[2] Information about the OpenAI API can be found here: `https://openai.com/blog/openai-api`

few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] S. Chasteen, K. Perkins, P. Beale, S. Pollock, and C. Wieman. A thoughtful approach to instruction: Course transformation for the rest of us. 2011.

[4] M. T. Chi, R. Glaser, and M. J. Farr. *The nature of expertise*. Psychology Press, 2014.

[5] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma. A survey on ensemble learning. *Frontiers of Computer Science*, 14:241–258, 2020.

[6] K. A. Ericsson, R. T. Krampe, and C. Tesch-Römer. The role of deliberate practice in the acquisition of expert performance. *Psychological review*, 100(3):363, 1993.

[7] T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. ROC Analysis in Pattern Recognition.

[8] P. Flowers and K. Theopold. [etextbook] chemistry-2e, 2019.

[9] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] G. Koch, R. Zemel, R. Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, page 0. Lille, 2015.

[11] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

[12] Y. Li, M. Rakovic, B. X. Poh, D. Gašević, and G. Chen. Automatic classification of learning objectives based on bloom's taxonomy. *International Educational Data Mining Society*, 2022.

[13] S. J. Ling, J. Sanny, W. Moebs, G. Friedman, S. D. Druger, A. Kolakowska, D. Anderson, D. Bowman, D. Demaree, E. Ginsberg, et al. University physics volume 2. 2016.

[14] A. Malik, M. Wu, V. Vasavada, J. Song, J. Mitchell, N. Goodman, and C. Piech. Generative grading: Neural approximate parsing for automated student feedback. *arXiv preprint arXiv:1905.09916*, 2019.

[15] E. G. Miller, N. E. Matsakis, and P. A. Viola. Learning from one example through shared densities on transforms. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 464–471. IEEE, 2000.

[16] J. E. Nardo. Ideal pedagogy stem presentation, 2021. Power Point Presentation, https://ideallabresearch.stanford.edu/.

[17] J. E. Nardo, N. C. Chapman, E. Y. Shi, C. Wieman, and S. Salehi. Perspectives on active learning: Challenges for equitable active learning implementation. *Journal of Chemical Education*, 99(4):1691–1699, 2022.

[18] R. Pepper, S. Chasteen, S. Pollock, and K. Perkins. Facilitating faculty conversations: Development of consensus learning goals. In *Physics Education Research Conference 2011*, volume 1413 of *PER Conference*, pages 291–294, Omaha, Nebraska, August 3-4 2011.

[19] A. Ramaprasad. On the definition of feedback. *Behavioral science*, 28(1):4–13, 1983.

[20] M. A. Ruiz-Primo, E. M. Furtak, C. Ayala, Y. Yin, and R. J. Shavelson. Formative assessment, motivation, and science learning. In *Handbook of formative assessment*, pages 139–158. Routledge, 2010.

[21] S. Salehi, E. Burkholder, G. P. Lepage, S. Pollock, and C. Wieman. Demographic gaps or preparation gaps?: The large impact of incoming preparation on performance of students in introductory physics. *Physical Review Physics Education Research*, 15(2):020114, 2019.

[22] S. Salehi, S. Cotner, and C. J. Ballen. Variation in incoming academic preparation: Consequences for minority and first-generation students. In *Frontiers in Education*, volume 5, page 552364. Frontiers Media SA, 2020.

[23] Y. Sasaki et al. The truth of the f-measure. *Teach tutor mater*, 1(5):1–5, 2007.

[24] L. A. Shepard. Classroom assessment to support teaching and learning. *The ANNALS of the American Academy of Political and Social Science*, 683(1):183–200, 2019.

[25] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[26] H. E. Tinsley and D. J. Weiss. Interrater reliability and agreement of subjective judgments. *Journal of Counseling Psychology*, 22(4):358, 1975.

[27] N. J. Tro. *Chemistry in focus: A molecular view of our world*. Cengage Learning, 2018.

[28] G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[30] M. Wu, R. L. Davis, B. W. Domingue, C. Piech, and N. Goodman. Variational item response theory: Fast, accurate, and expressive. *arXiv preprint arXiv:2002.00276*, 2020.

[31] M. Wu, N. Goodman, C. Piech, and C. Finn. Prototransformer: A meta-learning approach to providing student feedback. *arXiv preprint arXiv:2107.14035*, 2021.

APPENDIX

A. SAMPLE DATA

In this section we provide an overview of the OpenStax portion of our benchmark. Table 4 provides example input questions and their corresponding learning objective labels, sampled from our OpenStax training dataset.

We note that all questions from each sub-chapter of a given OpenStax textbook were labeled with every learning objective that the authors included for that subsection, rather than each question being hand-labeled with unique learning objectives. For example, all questions from the OpenStax Chemistry 2e [8] sub-chapter "6.1 Solving Problems with Newton's Laws" would be labeled with all five of the learning objectives for this sub-chapter (see Table 4). This simplifying assumption allowed us to create a much larger dataset, including 4,875 labeled questions spanning 1,267 specific learning objectives from OpenStax university-level science textbooks, than would have been possible had we hand-labeled each question individually. While not every question in each sub-chapter focuses on every learning objective for that sub-chapter, the key learning objectives for each question are likely to be included in the learning objectives for that question's sub-chapter. Similarly, it is likely that all of a sub-chapter's learning objectives are relevant in varying degrees to the questions from that sub-chapter, so questions from the OpenStax portion of our dataset are unlikely to be labeled with off-topic learning objectives.

The most significant limitation resulting from this simplifying assumption is that questions in our OpenStax dataset are never labeled with subsidiary learning objectives from other sub-chapters. In theory, this could limit the usefulness of the OpenStax portion of our dataset in training classifiers to label questions that focus on integrating multiple course topics. As shown by our experimental results (see Table 3), classifiers trained on our OpenStax dataset were able to perform effectively on our Chem 31A dataset, where chemistry experts individually hand-labeled questions with learning objectives. Additionally, many questions from the Chem 31A dataset focus on integrating skills from multiple course topics, particularly the longer free-response questions and questions from the final exam. Our classifiers' ability to generalize to the Chem 31A dataset after being trained on the OpenStax dataset suggests that the benefits of the method we used to label the OpenStax questions, such as enabling the creation of a much larger dataset for training, outweigh the limitations mentioned above.

B. SAMPLE OUTPUTS

Table 5 presents the outputs of the ProtoTransformer classifier with $k = 5$ on a sample of questions from the held-out Chem 31A course.

A brief inspection suggests that the ProtoTransformer classifier does not solely rely on semantic keywords. For example, although the second question contains the phrase "vapor pressure" multiple times, the top three classifier predictions do not contain this phrase. Meanwhile, the first question does not explicitly state the ideal gas law, $PV = nRT$; however, the classifier infers the learning objective label.

Although a thorougher investigation is required to interpret the ProtoTransformer classifier's behavior, we hypothesize that the classifier more accurately identifies core learning objectives (e.g. "use the ideal gas law", "interpret a phase diagram") which appear in many course questions, and less accurately predicts learning objectives that are specific to a sub-unit (e.g. "apply the concept of percent by mass"). This is because few-shot classification requires access to high-quality samples of related questions. Since the pool of questions related to the ideal gas law in Chem 31A is richer than the pool of questions related to the concept of percent by mass, the ProtoTransformer classifier is likely to achieve higher accuracy on the former than on the latter.

Table 4: Sample questions and their corresponding learning objective labels from the OpenStax training dataset.

Course + *subchapter*	Question	Learning Objectives
University Physics I *6.1 Solving Problems with Newton's Laws*	A 30.0-kg girl in a swing is pushed to one side and held at rest by a horizontal force F so that the swing ropes are 30.0° with respect to the vertical. (a) Calculate the tension in each of the two ropes supporting the swing under these conditions. (b) Calculate the magnitude of F	Apply problem-solving techniques to solve for quantities in more complex systems of forces Use concepts from kinematics to solve problems using Newton's laws of motion Solve more complex equilibrium problems Solve more complex acceleration problems Apply calculus to more advanced dynamics problems
Chemistry 2e *4.3 Reaction Stoichiometry*	What mass of silver oxide, Ag_2O, is required to produce 25.0 g of silver sulfadiazine, $AgC_{10}H_9N_4SO_2$, from the reaction of silver oxide and sulfadiazine? 2 $C_{10}H_{10}N_4SO_2$ + $Ag_2O \rightarrow$ 2 $AgC_{10}H_9N_4SO_2$ + H_2O	Explain the concept of stoichiometry as it pertains to chemical reactions Use balanced chemical equations to derive stoichiometric factors relating amounts of reactants and products Perform stoichiometric calculations involving mass, moles, and solution molarity

Table 5: Presents the outputs of the ProtoTransformer classifier with $k = 5$, run on four sample questions from the Chem 31A course. The top $m = 3$ learning objectives predicted by the classifier are shown for each question, in order of model confidence. Correct predictions by the model are highlighted in green, while incorrect predictions are highlighted in red.

Question	True Learning Objectives	Predicted Learning Objectives ($m = 3$)
A mixture of 20.0 g of Ne and 20.0 g Ar have a total pressure of 1.60 atm and temperature of 298K. What is the partial pressure of Ar?	Apply the concept of percent by mass and percent by volume when solving problems Use the ideal gas law (PV=nRT) to solve problems	🟥 Use gas laws with stoichiometry to analyze chemical reactions of gasses 🟩 Use the ideal gas law (PV=nRT) to solve problems 🟥 Write and balance chemical and net-ionic equations
Decreasing the external pressure on a liquid at constant temperature will do which of the following:(a) Increase the boiling point, but not affect the vapor pressure(b) Decrease the boiling point, but not affect the vapor pressure(c) Increase the vapor pressure, therefore decreasing the boiling point(d) Increase the amount of heat required to boil a mole of the liquid(e) Both B and D are true	Calculate how vapor pressure will change as the pressure, volume, temperature, or amount are varied Interpret a phase diagram to determine what phase change may occur for a given change in pressure or temperature	🟥 Calculate changes in energy, enthalpy, and temperature that result from a chemical reaction 🟩 Interpret a phase diagram to determine what phase change may occur for a given change in pressure or temperature 🟥 Know the difference between systems and surroundings
At a constant external pressure, if work was done by the system on the surroundings, would you expect ΔE for the system to be greater than, less than or the same as the ΔH° for the system?(a) ΔE for the system would be greater than ΔH°(b) ΔE for the system would be less than ΔH°(c) ΔE for the system would the same as ΔH°(d) It is impossible to determine without knowing the magnitude of work done.	Calculate the work done by or on a gas	🟥 Calculate how vapor pressure will change as the pressure, volume, temperature, or amount are varied 🟥 Know the difference between systems and surroundings 🟥 Use the ideal gas law (PV=nRT) to solve problems
Determine the longest wavelength of light capable of removing an electron from a sample of potassium metal, if the binding energy for an electron in K is 1.76 × 103 kJ/mol. (a) 147 nm (b) 68.0 nm (c) 113 nm (d) 885 nm (e) 387 nm	Know how the photoelectric effect can be used to assess binding energy Use the relationship between the frequency and wavelength and velocity (speed) of a wave to calculate any one (frequency, wavelength or velocity) given the other two	🟩 Know how the photoelectric effect can be used to assess binding energy 🟩 Use the relationship between the frequency and wavelength and velocity (speed) of a wave to calculate any one (frequency, wavelength or velocity) given the other two 🟥 Explain how electronic structure gives rise to periodic trends (i.e. recognizing isoelectronic species)

Clustering to define interview participants for analyzing student feedback: a case of Legends of Learning

Ayaz Karimov
Faculty of Information Technology
University of Jyväskylä
akarimov@jyu.fi

Mirka Saarela
Faculty of Information Technology
University of Jyväskylä
mirka.saarela@jyu.fi

Tommi Kärkkäinen
Faculty of Information Technology
University of Jyväskylä
tommi.karkkainen@jyu.fi

ABSTRACT

Within the last decade, different educational data mining techniques, particularly quantitative methods such as clustering, and regression analysis are widely used to analyze the data from educational games. In this research, we implemented a quantitative data mining technique (clustering) to further investigate students' feedback. Students played educational games within a week on the educational games platform, Legends of Learning and after a week, we asked them to fulfill the feedback survey about their feelings on the use of this platform. To analyze the collected data from students, firstly, we prepared clusters and selected one prototype student closest to the centroid of each cluster to interview. Interviews were held to explain the clusters more and due to time and resource limitations, we were unable to interview all (N=60) students, thus only the most representative students were interviewed. In addition to the students, we conducted an interview with the teacher as well to get her detailed feedback and observations on the usage of educational games. We also asked students to take an exam before and after the research to see the impact of games on their grades. Our results depict that though educational games can increase students' motivation, they may negatively impact some students' grades. And even though playing games made students feel interested and fun, they would not like to play them on a daily basis. Hence, using educational games for a certain duration such as subject revision weeks may positively influence students' grades and motivation.

Keywords

clustering, educational games, educational technology, serious game analytics, legends of learning

1. INTRODUCTION

Serious games or educational games can be defined as games that have the primary purpose of learning and education rather than entertainment. Educational games are special kinds of games that particularly aim to reach another outcome in addition to entertaining players. These games are used as a tool to increase the motivation and attention of students. In addition to the positive impact of educational games on students' attitudes, they can also directly help learners to increase their grades [31, 46]. However, it is important to mention that if not developed and implemented correctly, educational games can also negatively interfere with the learning outcome [22].

The implementation fields of educational games vary a lot. For example, while educational games were used in social science education such as English language education [1], history education [25], they were also widely used in more technical fields as well such as biology education [10], computer science education [30]. Particularly, within the teaching of complex subjects, educational games can help both teachers and students to ease the learning process. These games have the potential to get the attention of learners for a longer period compared to traditional lectures and by using different game elements, they encourage players to continue studying. [47] carried out the research about the systematic literature review on the use of educational games. Based on their analysis of published papers between 2009 and 2018, many factors can influence successful educational game usage. Gaming easiness, backstory, and production can be examples of these factors. Furthermore, in this research, we used the digital educational games platform called "Legends of Learning". Legends of Learning [1] is an online educational game platform that offers over 2,000 math and science games. Inside the platform, there are various games for each subject and class. Teachers create playlists of games based on standards and students work through completing each one.

The aim of this research was twofold:

- First, our goal was to measure the impact of utilizing educational games during science subject revision in one school in Azerbaijan. At the beginning and at the end of the revision week, students did two different tests: one pre and one postrevision test, which allowed us to measure the impact of educational games on their grades. Once the revision week ended, we asked students to provide feedback on the usage of educational games. Based on their test results and feedback on the

A. Karimov, M. Saarela, and T. Kärkkäinen. Clustering to define interview participants for analyzing student feedback: a case of legends of learning. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 234–243, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115667

[1] www.legendsoflearning.com

games, we developed four student clusters.

- Second, we aimed to implement a novel approach to the selection of interview participants rather than selecting them randomly. Because student clusters should be interpretable and representative of the whole student sample [37, 36]. While traditional educational data mining is more about quantitative analysis of educational data, qualitative data can give us a deeper understanding of individual students and their traits. Thus, the second aim of this research was to further refine the student clusters through qualitative analysis (semi-structured interviews) of each prototype student in the data (i.e., for each cluster, the student closest to the centroid).

2. LITERATURE REVIEW

2.1 Educational games in science education

Much research has been conducted on the usage of physical educational games in science education [27, 39, 8]. [23] developed an educational card game, then measured its impact on students' performance. They found that there is a significant difference between students' test results before and after playing this card game. Students also provided very satisfactory feedback to the researchers (4.29 over 5.00). Moreover, [6] found that using physical educational card games can enhance the educational experience of pharmacy students. [6] asked students to answer 90 questions before starting to play games and they played each game for 1 hour, 3 times over a 6-week period. The main subject of the games was cardiology and infectious diseases and students improved their assessment scores significantly (19.2% vs. 5.1%, ($p < 0.001$) and 10.3% vs. 5.1% ($p = 0.006$). In addition to increasing their grades, students also mentioned that they would like to play these card games in the future. Furthermore, [12] and [32] investigated educational board games. [12] used a board game called "Gut Check" where players try to develop a healthy microbiome for themselves and they also disrupt opponents' efforts. In the development of this game, the researchers worked with gamers and biologists to develop both educational and entertaining games. While [12] only focused on the educational board game, [32] both created a new educational board game whose structure of spatial relationships mirrored the structure of rational numbers and to measure the impact of the game, they implemented pretest-posttest assessments. [32] found that the correlation between posttest and pretest scores was not statistically significant, $r = .11$, $t(36) = 0.63$, $p = .531$. The baseline knowledge of their participants did not influence the estimated normalized knowledge gains. Even though these papers build a strong understanding and impact of using education in science education, they focus on physical educational games. In our research, we focused on the impact of digital educational games.

Some researchers investigated digital educational games and their impact on students' learning [42, 40, 5, 4]. [3] investigated the impact of using video gaming technology on middle school students learning within the scope of basic electromagnetism. For this, they used the game called "Supercharged!" which is a 3D action/racing game. In this game, players try to maneuver through a set of obstacles to obtain a certain goal. Supercharged! is designed based on the laws of electrostatics and the game helps the players to build stronger intuitions about how charged particles interact with electric and magnetic fields. [3] divided participants into two categories: experiment and control. Experiment group members learned the given physics subjects by playing games on Supercharged!, however, control group members learned the same physics subject by using the traditional learning method. The researchers found that the experiment group outperformed the control group and there was no significant difference from the perspective of gender. The researchers also asked open-ended questions to students and students mentioned that talking in the classroom during the learning process is not familiar to them, thus in the beginning it was challenging for some students to adapt. Additionally, based on their findings, digital educational games do not replace instruction, but they can support teaching. Moreover, [44] and [10] focused on the utilization of digital educational games in chemistry and biology education respectively. Both of these authors implemented pretest and posttest research methods where they divided students into experiment and control groups and asked students to take a test before and after playing digital educational games. [44] found that compared to the traditional teaching approach, the game-based learning approaches depicted better effects. Additionally, they also found that students are prone to have higher self-efficacy than those in a traditional lecture class when learning science. [10] found similar positive results from their research that there was a significant improvement in the overall learning achievement of students after playing digital games. Hereby, there was detailed research about the implementation of digital educational games in science education, nonetheless, we could not find research carried out in Azerbaijan. In our research, we focused on the usage of digital educational games in a school located in Azerbaijan.

Furthermore, in some cases, the implementation of educational games is not successful [47]. For instance, [13] and [15] found that educational games have a negative influence on the relationship between mental workload and learning effect. Additionally, [14] also investigated the digital educational games and they found that there were no significant differences in in-depth learning among learners.

2.2 Research on Legends of Learning

In this research, we used Legends of Learning as a digital educational game platform and we also did a literature review of the research carried out about the platform. [18] used the Legends of Learning platform to see how the platform impacts students' knowledge of the physics of light. 50 8th-grade students participated in the research and the pretest-posttest method was implemented. They found that there was a medium development for the concept mastery enhancement and student curiosity enhancement has shown a negative impact. Furthermore, the Legends of Learning team partnered up with the team from Vanderbilt University and they investigated the usage of educational games in the classroom [11]. They found that the students who used educational games as part of their regular curricula perform better than their peers on both factual knowledge and depth of knowledge. Based on our search, Legends of Learning was not utilized as a digital educational game platform in Azerbaijan, and in our research, we used this platform to see its impact on students' learning and motivation in a school located in Azerbaijan.

2.3 Educational games in Azerbaijan

In Azerbaijan, educational games and gamified apps are not familiar to the local market and the market lacks localized or translated international educational games platforms [26]. In [26], the researchers designed and developed an educational game called "FunMath" in which players need to solve mathematical problems to advance their scores. The app is designed from scratch in collaboration with the teachers and students. And after doing the usability test and final interview with the school teacher, they found that even though there could be more improvements, FunMath can extensively be used within and outside of school to increase learners' motivation.

[43] investigated how gamified environment can impact learners' motivation and math abilities. For this, they utilized the platform called "Polyup". Polyup is a computational thinking playground where students can experiment with numbers and functions. After playing games, they asked students to provide feedback by fulfilling the survey. Students mentioned that the platform can be used to connect mathematics with real-life experiences as well as can also enhance their mathematical calculation skills. Furthermore, based on our research, apart from [26] and [43], we could not find research focusing on educational games or gamified platforms in Azerbaijan. Additionally, both of these papers were focused on researching educational games within the scope of math education, and in our research, we focus on the impact of educational games in science education.

2.4 Educational games analytics by using clustering algorithms

Educational or serious games analytics refers to analytics or insights converted from gameplay data within educational games for the aim of performance measurement, assessment, or improvement [28]. Different supervised and unsupervised data mining methods can be implemented to analyze data from educational games [19]. Clustering is one of the unsupervised techniques of data mining and it contains various algorithms such as K-means, hierarchical clustering, and expectation maximization [38]. [2] did the systematic literature review study on the applications of data science techniques to analyze data from educational games. In their research, [2] found that 16 of the total 87 academic papers utilized clustering as a data mining technique which was one of the mostly-used methods along with linear regression and correlation analysis. And in this research, we also used the K-means algorithm to develop clusters.

[29] proposed and validated the questionnaire which is an instrument to measure the game preferences and habits of an intended audience. This instrument is called "The Game Preferences Questionnaire (GPQ) which possesses 10 Likert-scale items and produces a classification of the participants into four discrete clusters. While creating these clusters, [29] utilized the K-means algorithm and they found four main clusters: casual players, no gamers, well-rounded gamers, and hardcore gamers. Furthermore, [33] implemented K-means clustering, K-means++ initialization, and CVIs algorithms to investigate the creation of the new clustering-based profiling method. For this, they used the platform called "GraphoLearn" where users can play games for learning to read. [33] found that by utilizing their clustering method it is possible to cluster various kinds of learners and the method can help to track students who have reading disabilities. Additionally, [9] also used the K-means algorithm to analyze the measure of educational games on students. They used the platform called "OMEGA+" and in that platform, learners can play games to enhance their knowledge of problem-solving, planning and organization, associative reasoning, and accuracy and evaluation. After the implementation of the k-means algorithm, they found four clusters based on their activity status. [9] also mentioned that female players do not benefit from educational games due to their low activity status. In our research, in addition to the clustering analysis and interpretation of clusters, we utilized clustering results to select the students for the qualitative data collection. From this perspective, our paper brings novelty since it is a natural combination of qualitative and quantitative data analysis rather than only utilizing quantitative clustering data for quantitative data analysis purposes.

In addition to the K-means, other clustering algorithms were also used to analyze data from educational games. For instance, [17] used the density-based spatial clustering of applications with noise (DBSCAN) algorithm to research students' behavior in educational games. Based on the DBSCAN, [17] propose a new method called "SPRING" that helps student profile modeling in educational games. Furthermore, [24] focused on the analysis of player strategies in educational games, and for this, they utilized hierarchical clustering. Within this study, they used "GrAZE" which is a puzzle-based game where learners can improve their algorithmic thinking by playing this game. [24] highlighted problem areas that can be fixed in the early design phase.

3. METHODOLOGY

We implemented a mixed methodology: quantitative (surveys) and qualitative (interviews). To measure the change in students' learning, we implemented the pretest-posttest method where we asked students to take a test before and after playing games. Additionally, we also asked students to fulfill a survey which helped us to analyze their feedback and define clusters according to their background and test results. Furthermore, we selected one prototype student closest to the centroid of each cluster to interview (Figure 1).

3.1 Research set up and participants

Before starting the research process, we prepared agreements to be signed by the school principal and students' parents. All the permissions were collected from the principal and parents two weeks prior to the research. Students whose parents did not sign the agreement also played games but their data was not collected in any form. In parallel with the collection of permission forms, the teacher selected educational games from the platform for each class (Appendix C). On Legends of Learning, there are many games and each game has a very different learning goal, thus it was necessary to define the games for each class in advance. Moreover, we also communicated with all parents and school administrators to ensure that there is an internet connection in each classroom and students have their devices with them.

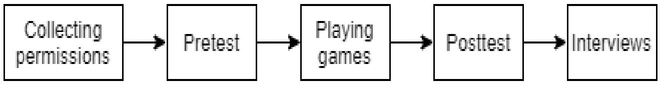


Figure 1: Research process

We recruited students who study in the 5th, 6th, or 7th grades(5th-grade students = 43.3%, 6th-grade students = 25%, 7th-grade students = 31.7%). We gathered data from 67 students, nevertheless, since 7 students could not take the test after playing educational games, we removed them from the dataset. Exactly 50% of students mentioned that they identify themselves as "male", and the remaining 50% selected the "female" option.

3.2 Legends of Learning platform

Legends of Learning contain many various games inside the platform and the type of the game type can differ from memory type to matching games. Thus, it totally depends on the learning goal of the instructor and what kinds of knowledge they want to deliver. In our research, since the teacher had prior experience using the platform in the classroom, she selected games (Appendix C). The selected games can mainly be categorized under three sections: matching, memory, and video games. In the matching games, students were asked to match images with the given information, and in the memory games, students were asked to memorize some of the notions. And in the case of video games, students play a regular game while in the middle of the game, the game stops and it asks the question from the student in a quiz format. If students answer questions correctly, then they collect points. Furthermore, one of the interesting features of Legends of Learning is the "Awakening" section. After finishing their regular games, students can move to the Awakening part, a virtual world. During the Awakening section, students can walk, meet each other, and solve additional questions. There is no endpoint in the Awakening section, thus one can collect as many points as one can. However, since we selected games previously and we wanted all of the students to play all selected games, we put a time limit of five minutes for the Awakening section.

3.3 Data collection and analysis

In this research, we collected both quantitative and qualitative data [2]. To collect quantitative data, we prepared the survey and arranged special dates for students to fulfill the survey at the school. To make sure that students understand how they need to fulfill the survey, the teacher explained each question to students while guiding them. In the development of the survey questions, we used closed-ended questions where participants choose one or more of the predetermined responses. The reason to select the closed-ended questions is that they are easier and faster to answer [45]. Considering the age of this study's participants, for us, it was necessary to keep their attention while they were fulfilling the survey. The main aim of this survey was to collect students' feedback. The feedback survey is designed in a five-item Likert scale ((1) Strongly disagree; (2) Disagree; (3) Neither agree nor disagree; (4) Agree; (5) Strongly agree). In the development of statements for the survey, we considered our main research focus such as what kind of information we would like students to provide, and additionally, we used [20] as a reference to select some of the statements. [20] developed these statements particularly to measure user feedback after using the e-learning games.

To measure the impact of educational games on the learning outcome, we implemented a pretest-posttest design where we asked students to take two tests: pregame and postgame tests [16]. The aim of asking students to take two tests was to investigate how students' grades changed before and after playing educational games. The difficulty of questions in both tests was very similar and the teacher prepared all questions. Moreover, on the first day of research, all students were asked to take a test (pregame test) for 20 minutes, and on the last of the research, students were again asked to take another test (postgame test). For the postgame test, students were also given 20 minutes to finish it. Furthermore, we wanted to measure whether the difference between pregame and postgame scores is significant or not. Since we had only two variables (pregame and postgame score), we conducted a t-test.

To create clusters, we implemented the K-means algorithm and we preprocessed the data before starting the algorithm implementation phase. Initially, since we did not have any missing values, we moved to the data transformation stage where we converted categorical variables into binary or numerical variables. As a next step, we standardized data by using MinMaxScaler. Within the implementation of the K-means algorithm, one of the important phases is to define the optimal number of clusters (k) and for this, we used the "Elbow method". When we visualized the graph we observed that the graph rapidly changed at a point and this happened when the number of K was 4. Furthermore, for clustering data, we used the following variables: students' responses to all questions in the feedback survey, pretest-posttest scores, gender, previous gaming experience, and grade that they study.

To collect qualitative data, we held semi-structured interviews. The main goal of holding these interviews was to collect detailed feedback from students and teachers about their experience using the platform. The reason why we selected semi-structured interviews was that we wanted to ask certain questions but we also wanted to investigate students' additional thoughts. While holding semi-structured interviews, it is possible to direct the interview based on participants' responses [35]. Furthermore, once we had the results from the clustering analysis, we selected one person from each cluster (the closest prototype to the centroid) because this student prototype would be the best representation of the cluster that they belong to. All interviews were held in the school based on the availability of the selected students and teachers. Interviews were recorded and transcribed in the Azerbaijani language, however, for this academic research, the main outputs were translated from Azerbaijani into English. The parents of the interview participants signed the agreement to give permission before we start interviewing. To analyze interview data, we implemented the coding scheme methodology [7].

[2] The datasets generated during and/or analyzed during the current study are available from the corresponding author on request.

Table 1: Overview of clusters

Cluster ID	Cluster description	N	Gender (female vs male)	Gaming experience (yes vs no)
cluster_0	Lowly motivated and positive-grade-change students	8	37.5% vs 62.5%	85.7% vs 14.3%
cluster_1	Average motivated and negative-grade-change students	28	50% vs 50%	85.7% vs 14.3%
cluster_2	Average motivated and no-grade-change students	16	43.8% vs 56.2%	75% vs 25%
cluster_3	Highly motivated and high-positive-grade students	9	67% vs 33%	100% vs 0%

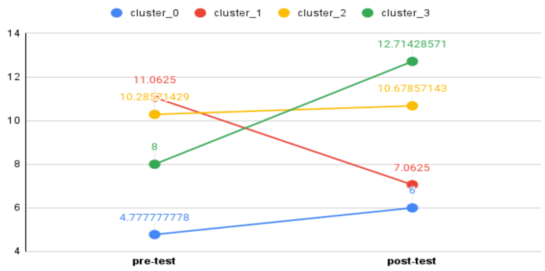


Figure 2: Change in the grades of each cluster on average

4. RESULTS

4.1 Survey results

In the survey, students provided feedback by answering questions where the maximum score was 5 and the minimum score was 1. When we asked students about their feeling while they revise the subject by playing games, they mentioned that they felt more satisfied (4.07) and more motivated (3.67). Additionally, students gave a score of 4.22 for their experience from the perspective of enjoyment. It depicts that students did not feel bored or anxious while playing these games. Subsequently, students mentioned that they would like to revise the study subjects by playing educational games in the future as well (4.32). However, students gave a score of 3.65 to the statement about improvement in their knowledge. To sum up, even though students provided high scores to the statements about their feelings and willingness to use educational games in the future, their scores for knowledge did not improve significantly.

4.2 Student profiles based on clustering results

According to the clustering results, we made four different student profiles (Table 1) and they are as follows:

1. Highly motivated and high-positive-grade students: Students in this cluster changed their grades by 4.71 points on average (Figure 2). They were satisfied with their experience on the platform and according to them, educational games helped them to increase their knowledge. Lastly, these students enjoyed the games most.

2. Average motivated and no-grade-change students: These students felt less motivated and they felt bored more at some parts of the games compared to the previous cluster of students. Moreover, there was no change in the grades of these students after playing educational games.

3. Average motivated and negative-grade-change students: The main characteristics of these cluster students look alike with the second cluster students from the perspective of how they feel using educational games. The main difference between the second and this cluster was their grade change. Students in this cluster decreased their grades by 4 points on average. This was the only cluster that depicted a negative change in their grades after playing educational games.

4. Lowly motivated and positive-grade-change students: Students in this cluster felt less motivated and they were not satisfied in comparison with their peers. Even though they provided lower points for their feelings about using the platform, these students made a positive change in their grades by 1.3 points on average.

When we ran a t-test to measure the significance of the change, we found that there is no sufficient evidence to say that the average grade of students before and after using educational games is different (p-value = 0.7746, alpha = 0.05). Moreover, we also analyzed the average score of each cluster to the feedback survey where “5” means that they strongly agree with the statement and “1” signifies that they strongly disagree with the statement. Overall, cluster_3 provided very positive feedback from any perspective such as being motivated to use educational games in the future and feeling positive and joyful while playing games. Cluster_0 provided lower scores compared to other clusters on average and the scores from cluster_1 and cluster_2 were almost the same. Nevertheless, almost all clusters (except cluster_2) gave the lowest score to the statement about the increase in their knowledge (“The games increased my knowledge.”).

4.3 Students’ and teacher’s feedback on the platform

To collect detailed feedback, we selected one prototype student closest to the centroid of each cluster. During the interviews, students provided their feedback about the platform and their educational games experience (Appendix A). Most of the students highlighted positive points about the usage of educational games and they also emphasized that Legends of Learning is a very user-friendly platform, thus it was a good experience to play games on this platform (cluster_3 and cluster_2). Moreover, all interviewed students mentioned that they own previous experience in using gamified educational tools or educational games. Students also provided feedback about the disadvantages of using educational games. Cluster_2 and cluster_3 prototype students mentioned that it is challenging for them to learn new knowledge on the platform. Even though they encountered new notions and terms, after playing games, they could not remember most of them. Furthermore, cluster_1 prototype student noted that in some parts of the game, answering questions correctly did not influence their game performance, so they were trying to click any buttons so that they can move to the next stage faster. Lastly, cluster_0 prototype students mentioned that some types of games such as memory games even decreased their motivation to continue.

In addition to students, we also conducted an interview

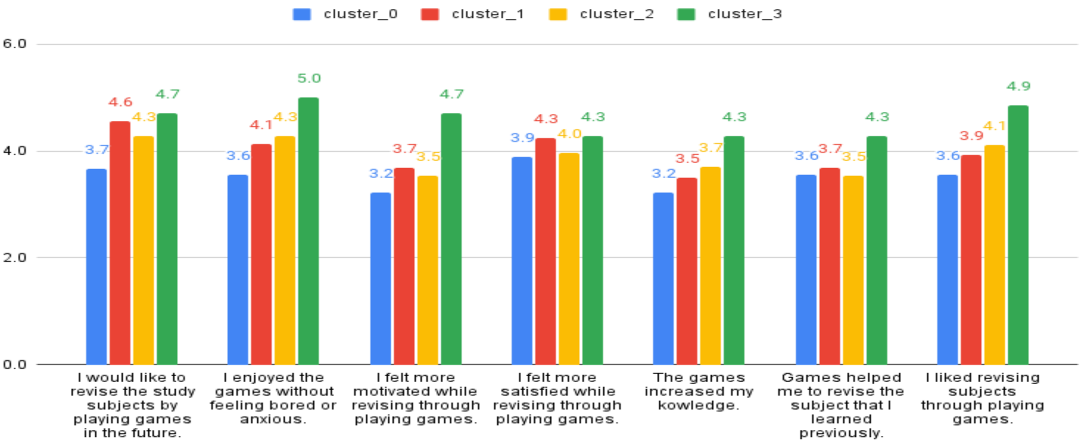


Figure 3: Responses of each cluster to the survey questions

with the teacher to understand students' attitudes, her feedback on teaching through educational games, and the challenges that she encountered during the research week (Appendix B). Overall, the teacher mentioned positive feedback on the use of educational games since she observed a positive change in students' behavior. She highlighted that students collaborated with one another to explain some problems and it increased the engagement inside the classroom. Particularly, after finishing the research, some inactive students started to participate in the classroom more actively and she thought that games helped them to see science from a different perspective. The teacher also mentioned some challenges and negative feedback that she observed and heard from students. According to her, some students did not have any prior knowledge of playing games or even using computers, thus adoption of the educational games took longer time compared to their peers. Moreover, even though playing educational games was fun, students were not interested in playing them on a daily basis.

5. DISCUSSION AND CONCLUSION

The objective of the present study was to investigate the impact of utilizing educational games as a tool to revise science subjects. In the first step, we asked students to take a test before starting to play educational games. After taking a pregame test, they played games on the platform called "Legends of Learning" for a week. On the last day of the week, students took a postgame test. Pregame and postgame tests helped us to measure the change in students' grades. Subsequently, students fulfilled a feedback survey where we asked them to rate statements about their feeling about using educational games. Based on the data from the feedback survey and their pregame and postgame test scores, we created clusters by using the K-means clustering algorithm. Then, we selected one prototype student from each cluster to get detailed feedback on their experience of using educational games. As a last step, we did an interview with a teacher to understand her perspective on teaching through educational games and students' behavior change based on her observation.

[34] and [21] highlight that players with prior gaming experience positively impact their performance in other games. In our research, we also found a similar relationship between students' prior gaming experience and their attitudes. Most of the students had prior experience playing educational games or using gamified platforms and all students mentioned that playing games in the classroom was entertaining (cluster_0), exciting (cluster_1), fun (cluster_2), and interesting (cluster_3). [3] mentioned that educational games cannot replace traditional instructions, however, they can support learning. Based on the interviews with students, we can also see similar answers where they highlighted encountering difficulties to learn a new subject by only playing games. All students and the teacher mentioned that they would prefer playing games only during revision weeks since regular lectures help them to learn more effectively. Additionally, as [41] found in their papers, the teacher also mentioned that games motivated introverted and passive students and they started participating in classroom discussions.

There are limitations to this work that should be noted. In this research, we used one digital educational games platform and within this study, we focused on elementary school students which restricted the scope of the research. Due to our resources, it was only possible to hold this research with a group of students, and the implementation of one platform was possible. Moreover, there is a wide range of future work that we want to address based on the results presented in this paper. Firstly, the participant profile can be changed to see whether it affects their motivation and grades differently. For instance, holding this research with primary and secondary school students can result in different outcomes. Secondly, instead of utilizing Legends of Learning, another digital educational games platform can be used. Because each platform is different and may bring different results. Last not but least, the games inside the Legends of Learning can be further studied.

5.1 Ethical concerns

To maintain participants' confidentiality they were assigned a number rather than their name, and data were stored and will be disposed of securely according to the agreement that parents signed. Students were also given the right to withdraw whenever they want to stop and leave at any point in the study.

6. ACKNOWLEDGEMENT

We would like to acknowledge the Science teacher, Sabina Mammadova, and the administration of the IB MYP Programme of Educational Complex 132-134 for their invaluable support in data collection and setting up the research. Their assistance was essential to the successful completion of this study, and we are grateful for their dedication to advancing the field of education.

7. REFERENCES

[1] J. M. Al-Jarrah, O. T. Waari, R. H. Talafhah, and T. M. Al-Jarrah. Improving english grammar achievement through educational games among eleventh grade students in east jerusalem. *International Journal of Academic Research in Progressive Education and Development*, 8(1):75–86, 2019.

[2] C. Alonso-Fernández, A. Calvo-Morata, M. Freire, I. Martínez-Ortiz, and B. Fernández-Manjón. Applications of data science to game learning analytics data: A systematic literature review. *Computers & Education*, 141:103612, 2019.

[3] J. L. Anderson and M. Barnett. Learning physics with digital game simulations in middle school science. *Journal of science education and technology*, 22(6):914–926, 2013.

[4] J. W. J. Ang, Y. N. A. Ng, and R. S. Liew. Physical and digital educational escape room for teaching chemical bonding. *Journal of Chemical Education*, 97(9):2849–2856, 2020.

[5] L. A. Annetta, M.-T. Cheng, and S. Holmes. Assessing twenty-first century skills through a teacher created video game for high school biology students. *Research in Science & Technological Education*, 28(2):101–114, 2010.

[6] S. M. Barclay, M. N. Jeffres, and R. Bhakta. Educational card games to teach pharmacotherapeutics in an advanced pharmacy practice experience. *American Journal of Pharmaceutical Education*, 75(2):33, 2011.

[7] W. Barendregt and M. M. Bekker. Developing a coding scheme for detecting usability and fun problems in computer games for young children. *Behavior research methods*, 38(3):382–389, 2006.

[8] Y. Bouzid, M. A. Khenissi, F. Essalmi, and M. Jemni. Using educational games for sign language learning-a signwriting learning game: Case study. *Journal of Educational Technology & Society*, 19(1):129–141, 2016.

[9] D. Chandrasekaran, M. Chang, and S. Graf. A learning analytics approach to build learner profiles within the educational game omega+. In *International conference on intelligent tutoring systems*, pages 139–147. Springer, 2022.

[10] M.-T. Cheng, T. Su, W.-Y. Huang, and J.-H. Chen. An educational game for learning human immunology: What do students learn and how do they perceive? *British Journal of Educational Technology*, 45(5):820–833, 2014.

[11] D. B. Clark, E. Tanner-Smith, A. Hostetler, A. Fradkin, and V. Polikov. Substantial integration of typical educational games into extended curricula. *Journal of the Learning Sciences*, 27(2):265–318, 2018.

[12] D. A. Coil, C. L. Ettinger, and J. A. Eisen. Gut check: The evolution of an educational board game. *PLoS Biology*, 15(4):e2001984, 2017.

[13] B. Cowley, M. Fantato, C. Jennett, M. Ruskov, and N. Ravaja. Learning when serious: Psychophysiological evaluation of a technology-enhanced learning game. *Journal of Educational Technology & Society*, 17(1):3–16, 2014.

[14] B. Cowley, T. Heikura, and N. Ravaja. Learning loops–interactions between guided reflection and experience-based learning in a serious game activity. *Journal of Computer Assisted Learning*, 29(4):348–370, 2013.

[15] B. Cowley, N. Ravaja, and T. Heikura. Cardiovascular physiology predicts learning effects in a serious game activity. *Computers & Education*, 60(1):299–309, 2013.

[16] D. M. Dimitrov and P. D. Rumrill Jr. Pretest-posttest designs and measurement of change. *Work*, 20(2):159–165, 2003.

[17] M. H. Falakmasir, J. P. González-Brenes, G. J. Gordon, and K. E. DiCerbo. A data-driven approach for inferring student proficiency from game activity logs. In *Proceedings of the third (2016) acm conference on learning@ scale*, pages 341–349, 2016.

[18] A. F. N. Fikriah. *Addressing Concept Mastery and Curiosity about the Physics of Light in Middle School Students through Discovery Learning with "Legends of Learning" Educational Games*. PhD thesis, Universitas Pendidikan Indonesia, 2021.

[19] M. Freire, A. Serrano-Laguna, B. Manero, I. Martinez-Ortiz, P. Moreno-Ger, and B. Fernandez-Manjon. *Game Learning Analytics: Learning Analytics for Serious Games*, pages 1–29. Springer Nature Switzerland AG, Switzerland, Apr. 2016.

[20] F.-L. Fu, R.-C. Su, and S.-C. Yu. Egameflow: A scale to measure learners' enjoyment of e-learning games. *Computers & Education*, 52(1):101–112, 2009.

[21] V. Garneli, M. Giannakos, and K. Chorianopoulos. Serious games as a malleable learning medium: The effects of narrative, gameplay, and making on students' performance and attitudes. *British Journal of Educational Technology*, 48(3):842–859, 2017.

[22] M. Gaydos. Seriously considering design in educational games. *Educational Researcher*, 44(9):478–483, 2015.

[23] A. F. Gutierrez. Development and effectiveness of an educational card game as supplementary material in understanding selected topics in biology. *CBE—Life Sciences Education*, 13(1):76–82, 2014.

[24] B. Horn, A. K. Hoover, J. Barnes, Y. Folajimi, G. Smith, and C. Harteveld. Opening the black box of play: Strategy analysis of an educational game. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, pages 142–153, 2016.

[25] J. Huizenga, W. Admiraal, G. Ten Dam, and J. Voogt. Mobile game-based learning in secondary education: Students' immersion, game activities, team

performance and learning outcomes. *Computers in Human Behavior*, 99:137–143, 2019.

[26] S. Huseynli. Application of games in teaching mathematics at primary and preschools. 2021.

[27] G. Jin, M. Tu, T.-H. Kim, J. Heffron, and J. White. Game based cybersecurity training for high school students. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, pages 68–73, 2018.

[28] J. Kang, M. Liu, and W. Qu. Using gameplay data to examine learning behavior patterns in a serious game. *Computers in Human Behavior*, 72:757–770, 2017.

[29] B. Manero, J. Torrente, M. Freire, and B. Fernández-Manjón. An instrument to build a gamer clustering framework according to gaming preferences and habits. *Computers in Human Behavior*, 62:353–363, 2016.

[30] R. Mathew, S. I. Malik, and R. M. Tawafak. Teaching problem solving skills using an educational game in a computer programming course. *Informatics in education*, 18(2):359–373, 2019.

[31] U. Munz, P. Schumm, A. Wiesebrock, and F. Allgower. Motivation and learning progress through educational games. *IEEE Transactions on Industrial Electronics*, 54(6):3141–3144, 2007.

[32] J. A. Navarrete-Ulloa and F. Munoz-Rubke. Playing board games to learn rational numbers: A proof-of-concept. *Mind, Brain, and Education*, 16(4):293–299, 2022.

[33] M. Niemelä, T. Kärkkäinen, S. Äyrämö, M. Ronimus, U. Richardson, and H. Lyytinen. Game learning analytics for understanding reading skills in transparent writing system. *British Journal of Educational Technology*, 51(6):2376–2390, 2020.

[34] C. O'Donovan and J. Hussey. Active video games as a form of exercise and the effect of gaming experience: a preliminary study in healthy young adults. *Physiotherapy*, 98(3):205–210, 2012.

[35] S. E. Rabionet. How i learned to design and conduct semi-structured interviews: an ongoing and continuous journey. *Qualitative Report*, 16(2):563–566, 2011.

[36] M. Saarela, V. Heilala, P. Jääskelä, A. Rantakaulio, and T. Kärkkäinen. Explainable student agency analytics. *IEEE Access*, 9:137444–137459, 2021.

[37] M. Saarela, J. Hämäläinen, and T. Kärkkäinen. Feature ranking of large, robust, and weighted clustering result. *Lecture Notes in Computer Science*, 10234:96–109, 2017.

[38] M. Saarela and T. Kärkkäinen. Knowledge discovery from the programme for international student assessment. In *Learning Analytics: Fundaments, Applications, and Trends*, Studies in systems, decision and control, pages 229–267. Springer International Publishing, Cham, 2017.

[39] N. R. Scalise, E. N. Daubert, and G. B. Ramani. Benefits of playing numerical card games on head start children's mathematical skills. *The Journal of Experimental Education*, 88(2):200–220, 2020.

[40] P. Sengupta, K. D. Krinks, and D. B. Clark. Learning to deflect: Conceptual change in physics during digital game play. *The Journal of the Learning Sciences*, 24(4):638–674, 2015.

[41] R. Smiderle, S. J. Rigo, L. B. Marques, J. A. Peçanha de Miranda Coelho, and P. A. Jaques. The impact of gamification on students' learning, engagement and behavior based on their personality traits. *Smart Learning Environments*, 7(1):1–11, 2020.

[42] K. Squire, M. Barnett, J. M. Grant, and T. Higginbotham. Electromagnetism supercharged! learning physics with digital simulation games. 2004.

[43] Y. Tabesh, S. Zarkesh, A. Zarkesh, and I. Fazilova. Computational thinking in k-12: Azerbaijan's experience. *Olympiads in Informatics*, 13:217–224, 07 2019.

[44] M. Wang and X. Zheng. Using game-based learning to support learning science: A study with middle school students. *The Asia-Pacific Education Researcher*, 30(2):167–176, 2021.

[45] S. Yaddanapudi and L. Yaddanapudi. How to design a questionnaire. *Indian journal of anaesthesia*, 63(5):335–337, 2019.

[46] Z. Yu, M. Gao, and L. Wang. The effect of educational games on learning outcomes, student motivation, engagement and satisfaction. *Journal of Educational Computing Research*, 59(3):522–546, 2021.

[47] Y. Zhonggen. A meta-analysis of use of serious games in education over a decade. *International Journal of Computer Games Technology*, 2019:1–8, 2019.

APPENDIX

A. STUDENTS' FEEDBACK

Cluster_0 student:

> Since the teacher does not explain anything, it is very hard to actually understand anything from scratch. Because in the classroom, our teacher explains everything many times and very easily but in the games, they just provide information and the explanation is not detailed. So maybe I would not like to play games to learn something from scratch. And I was not sure whether I am learning new things or not and that was bad. I would not like to play games daily because I would feel bored after some time but for the revision week, it would be amazing to play games always. I liked playing games because I felt more entertained in the classroom. I play these kinds of games very often at my tutoring sessions and it always makes me want to play more. The games on Legends of Learning were okay and I did not feel that bored except for memory games. Because in the memory games, I struggled a lot and I could not find many hints to use. But in general, it was good and exciting to play these games in the classroom.

Cluster_1:

> It was neither good nor bad, it was fun to play games but after a certain point, I felt a bit bored. At the beginning of each class day, I was excited to play games and I think some of the games are not as fun as others. But I think the platform

is good because it does not have many bugs because most of the games that I play at home have bugs or they work very slowly. I especially liked the Awakening section of the platform. Because I liked that there are no limits and I could do many things. Actually, at that point, I understood that I can just keep playing even if I answer the question correctly or wrongly. And I also remember that I can skip some points in which I could learn something. So it was a bit bad that sometimes I did not feel I am learning anything I was only playing games and ignoring the learning part. However, in the classroom, our teacher generally asks random questions in the classroom and I try to follow her so that I can answer her question. I think we can use games only for the revision week and it would not be good to use them every day.

Cluster_2:

> I liked playing games during the revision week, it was very fun. In some parts of the game, I was asked to answer some questions and in these parts, I memorized what I learned previously in the class so it was very good to revise previous subjects. I also learned new information on the platform. Maybe we learned this in the classroom previously, I learned some of them in the games. Normally, I struggle a lot to sit and try to read something but by playing games, it felt more like free-time activity rather than really studying something. Even I was very excited to be in Science class. I previously played some educational games such as Kahoot! and Bluekit but they were only in the French and Biology classes. In general, I like playing educational games. I think the biggest disadvantage is that it is very hard to learn something new by only playing games. For example, I learned some new things but I already forget most of them. Because they appeared only in one part of the game and I used them to move to the next sessions. But I think that we can play games two or three times a month about the subjects that we already learned at school.

Cluster_3:

> It was so much fun and interesting to play these games. I try to find and play similar games at home as well and these ones were very entertaining. But at home, I start feeling bored after some time and the best thing was that I was playing games together with my friends at school. Honestly, it did not feel like a lesson at all. I think I also learned new things by playing games. It was also good to answer questions that I already learned at school. I think I collected many points because I was trying to remember and select the right answer to each question calmly. I liked games on Legends of Learning because they differ from one another a lot. For example, I did not like memory games and in these games, and I was trying to finish the memory games and move on to more interesting ones. I do not think it has a huge disadvantage because I really enjoyed but maybe it is only good for the revision week. Because I was mainly using my previous knowledge on the topic and that is why I managed to finish many games earlier than my friends. And I also want to say that I think some students were just clicking randomly just to continue and collect points in the knowledge-sharing parts.

B. NOTES FROM THE INTERVIEW WITH THE TEACHER

Teacher:

> Students enjoyed the platform and the Awakening was the most interesting part of the section because they were able to see one another and interact. In general, they were more motivated when they were collaborating and talking to one another. Some students whom we can call “a gamer” were more prone to solve problems and finish problems very quickly. Once they finished their games, they were trying to help their friends by giving hints and explaining the platform. There was also one interesting situation with one of my students who was very introverted and silent during our casual classes. But during the research week, I saw that she was very active and a pioneer to finish games way earlier than their peers. So I think educational games directly impact gamers who are not much active in regular classes. One of the greatest things was that there were at least six students who changed their attitudes toward the Science class as well. Before this research, these six students were not active in the classroom, but even after the research week, I can clearly see that they have more interest in Science class. I think they saw the different perspectives of Science and they really liked it. Students understand that the games are fun but they prefer the teacher to explain something and learn before. Because in some games, they particularly said that they did not learn this topic in the classroom so they could not pass to the next level. So I think these kinds of educational games are only good to revise some subjects. I also saw some students that were trying to skip the instructions part and maybe they managed to get some time in the beginning, but once they moved to the games part, they struggled a lot. Even some needed to go back and read the instructions again.

C. SELECTED EDUCATIONAL GAMES

The games for the 5th-grade students:

- Particle trip: Structure of matter
- Matter memory
- Attack of the ice giants
- Matter popper
- LAB fever

- Chemibot helps the city
- The roles of water in Earth's surface processes

The games for the 6th-grade students:

- Population Frenzy
- Weather master
- Tornado tournament
- Climate cities
- The water cycle
- Tectonic designers
- Seafloor explorer

The games for the 7th-grade students:

- The spark of life
- Dr. Franks' cell matching adventure
- Codex - neural disarray
- Cell explorers
- Ener-jump
- Little big plant
- Photosynth Adventure

The Predictiveness of PFA is Improved by Incorporating the Learner's Correct Response Time Fluctuation

Wei Chu and Philip I. Pavlik Jr.
University of Memphis
wchu, ppavlik@memphis.edu

ABSTRACT

In adaptive learning systems, various models are employed to obtain the optimal learning schedule and review for a specific learner. Models of learning are used to estimate the learner's current recall probability by incorporating features or predictors proposed by psychological theory or empirically relevant to learners' performance. Logistic regression for knowledge tracing has been used widely in modern learner performance modeling. Notably, the learning history included in such models is typically confined to learners' prior accuracy performance without paying attention to learners' response time (RT), such as the performance factors analysis (PFA) model. However, RT and accuracy may give us a more comprehensive picture of a learner's learning trajectory. For example, without considering RT, we cannot estimate whether the learner's performance has reached the automatic or fluent level since these criteria are not accuracy based. Therefore, in the current research, we propose and test new RT-related features to capture learners' correct RT fluctuations around their estimated ideal fluent RT. Our results indicate that the predictiveness of the standard PFA model can be increased by up to 10% for our test data after incorporating RT-related features, but the complexity of the question format constrains the improvement during practice. If the question is of low complexity and the observed accuracy of the learner can be influenced by guessing, which results in the imprecision measured by accuracy, then the RT-related features provide additional predictive power. In other words, RT-related features are informative when accuracy alone does not completely reflect learners' learning processes.

Keywords

Performance Factors Analysis, Response Time, Memory, Fact Learning, Logistic Regression

1. INTRODUCTION

As early as Atkinson [4], model-based adaptive scheduling has been explored extensively and deeply to improve learners' learning efficiency and long-term retention. According to the theory of knowledge tracing [9], one general and important preceding step behind this sort of research is to build a learner model that can accurately estimate the learner's probability of correctly answering questions they will encounter based on their prior behaviors [10]. One area of learner modeling methods is derived from Item Response Theory (IRT) framework, leveraging the Rasch model's logistic transformation [34]. Several different logistic regression learning models have been successfully built by considering different facets of learners' learning history, such as the Additive Factors Model (AFM) [7], which uses the number of prior practices, the Performance Factors Analysis (PFA) [32] which uses the performance (correct or incorrect) on previously practiced items, the Instructional Factors Analysis (IFA) [8] which uses the previous instructional interventions the learner has received, in addition to many other predictors reviewed recently [30].

We noticed that the response time (RT), one commonly used indicator in cognitive domains, was not used in such adaptive models, despite its long history as a factor that traced learning [15]. However, when depicting a learner's performance, accuracy is not enough to give us the whole picture of the learner's learning trajectory. Accuracy is discrete and may not be precise enough when measuring learners' learning. For example, the learner's incorrect responses could be caused by slipping, and similarly, the learners' correct responses could be caused by guessing [5]. Therefore, to measure learners' learning and performance more precisely, we hypothesize that RT and accuracy during learning should be used jointly. For example, quicker correct responses indicate learners have stronger memory traces of materials [1, 43]. Furthermore, responding fluently or automatically is often seen as a criterion of learning and training in practical situations [25], such as foreign language, emergency medicine, and simple facts learning [14, 18, 42], so incorporating it as a predictor may increase the generalizability of such modeling.

Considering the connection between learners' RT and their performance, some researchers have integrated information implied by RT in adaptive learner modeling [11, 37] and experimentally validated the effectiveness of such RT-based components in improving learners' acquisition and retention [19, 20, 22, 23, 24, 25, 38, 39, 40, 41, 42]. For instance, Sense and van Rijn [42] incorporated the learner's observed RT to adjust the model's parameter controlling the decay rate of a specific item and showed that RT is informative and can significantly contribute to predicting recall. Their results showed that the scheduling algorithm incorporating the RT information results in higher retention than the random presentation schedule. Similarly, Mettler and colleagues [25] assumed that compared to slow correct RT, faster correct RT for a specific item reflects the learner has stronger learning strength of the item. Thus, in their adaptive response time-based sequencing system (ARTS), items that have been answered correctly and quickly would be repeated in a longer recurrence interval for the learner. Consequently, the ARTS system outperforms the Atkinson [4] method in learning efficiency [24]. However, Lindsey et al. [19, 20] pointed out that despite the predictiveness power of learners' future performance provided by RT, it was redundant with information held in the accuracy. Thus, RT information of learners was not used in their later

W. Chu and P. I. P. Jr. The predictiveness of pfa is improved by incorporating the learner's correct response time fluctuation. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 244–250, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115643

adaptive scheduling system, DASH [22]. Table 1 briefly summarizes both adaptive scheduling systems incorporating RT information.

Table 1. Summary of RT-related features in different adaptive scheduling systems

System	Theoretical Assumption	Model Mechanic	RT-related features
Sense & van Rijn [42]	Strength theory: Correct response speed positively correlates to memory trace strength [27]	ACT-R declarative memory module [2, 33]	A parameter (α) represents the decay rate of memory traces
Mettler et al. [25]	Learning strength: A hypothetical construct related to probability of future successful recall	Adaptive Response Time-Based Sequencing (ARTS)	Priority Score for items that have been answered correctly

In summary, from a theoretical perspective, RT-related features are informative for capturing facets of individual differences, such as the memory strengths of items during practice. In practical applications, whether the prediction of learning models can be improved after incorporating RT-related features still needs further exploration due to the noisy nature of RT data in many applications. Therefore, in the current research, we investigated if the predictiveness of the standard PFA model can be improved by incorporating the learner's correct RT history. Specifically, we focused on estimating the learner's fluent RT after reaching the automatic response level, then compared the learner's correct RT during learning with their estimated fluent RT to capture the strength changes of memory traces.

2. METHOD

2.1 Performance Factors Analysis

Performance factors analysis (PFA) is a logistic regression model using learners' prior practice performance on knowledge components (KC) to estimate their future probability of a correction [32]. A KC is defined as a mental structure or process a learner uses alone or in combination with other KCs to solve problems [17] and can be operationalized as facts, concepts, or complex skills depending on the granularity of analyses. In PFA, the learner's performance, correct and incorrect responses are selected as indicators of learners' learning processes. The mathematical format for PFA is shown in Equation 1 and Equation 2. Equation 1 captures the strength values for KCs, where i represents an individual learner, j represents a specific KC, β represents the easiness of the KC, α represents the ability of the learner, s tracks the prior successes for the KC for the learner (γ scales the effect of these prior successes count), and f tracks the prior failures for the KC for the learner (ρ scales the effect of these prior failures counts). Equation 2 converts strength values to predictions of correctness probability according to the logistic distribution. Since the standard PFA does not integrate the information provided by learners' RT, which is also probably a strong indicator of learners' learning, we believed the modifications we conducted for the standard PFA described in the following sections would be helpful.

$$m(i,\ j \in KC_s,\ s,\ f) = \sum_{j \in KC_s} \left(\gamma_j s_{i,j} + \rho_j f_{i,j} + \beta_j\right) + \alpha_i \quad (1)$$

$$p(m) = (1 - e^{-m})^{-1} \qquad (2)$$

2.2 Variants of PFA with Correct-RT-Related Features

2.2.1 *The Exponential Law of Practice*

The "law of practice" function describes the relationship between RT and practice opportunities. Many researchers have shown that simple mathematical functions can fit this relationship [3, 13, 28]. Anderson [3] showed that RT is an exponential function of memory activation, and the intercept can capture a learner's neural integration time and motor response time. Newell and Rosenbloom [28] showed that RT follows a power function of prior practice opportunities. Heathcote and colleagues [13] extensively compared the overall fitting of exponential functions and the power functions across 40 sets of data, and they found that for unaveraged data, such as data from individual learners which were commonly used in adaptive modeling, the exponential function fitted the data better than the power function. As it turns out, averaging exponential functions produces power functions, making these results sensible [3].

Thus, in the present research, to fit the individual learner's RT as a function of the practice opportunity, we used the exponential function as shown in Equation 3, where $E\ (RT_n)$ represents the expected value of RT on practice opportunity n , B represents the change in the expected value of RT from the beginning of learning (n = 0) to the end of learning (the x_{th} practice opportunity when the learner reaches their fluent RT), A_i represents the expected value of RT after learning has been completed for the individual learner i , and α is the rate parameter and controls the amount of nonlinearity displayed by the exponential function.

$$E\ (RT_n) = A_i + Be^{-\alpha n} \quad (3)$$

Our main goal was to estimate the value of A_i for the individual learner i , which represents the RT needed for the learner to perform fluently (*fluent*$_{RT}$). In other words, we assumed that if the learner truly mastered the materials, no retrieval time would be included in A_i implying an automatic response that captures a learner's neural integration time and motor response time. The estimation was conducted using the *optim* function from the *'stats'* R package [35].

2.2.2 *Correct-RT-Related Features and PFA Variants*

After having the estimated *fluent*$_{RT}$ value of each learner, we need other correct RT information from the learner's practice history to calculate predictive components to examine whether incorporating such correct-RT-related features added to the standard PFA improves its predictiveness. We followed the method used by Eglington and Pavlik [10]. For each learner i , for each KC j , and each trial t , a median trial RT was calculated from the previous trials $1{:}\ t-1$ for which the learner was correctly answered. For the first trial for a specific learner, and all trials before a correct response had been produced, the value was set to zero (hereafter, this value was named *median*$_{corRT}$). A dummy variable (*dummy*) was also created and also added to the model. The dummy captures the performance difference between first trials and other consecutive wrong trials at the beginning of the practice session where calculating the *median*$_{corRT}$ is impossible. For example, suppose the learner's responses are (wrong, wrong, wrong, wrong, correct with latency 4000ms) for the first five trials for the same KC. In that case, the calculation for this learner's running *median$_{corRT}$* is (0, 0, 0, 0, 4000), where the corresponding *dummy* code for the learner's first five responses was (1, 1, 1, 1, 0). This *dummy* provides a baseline for all trials before the first correct result, which offsets the value of 0 that is needed to predict the correct latency effect (0 since there has been no correct latency). Since we cannot use 0 for these trials (since it is just a placeholder), we need this *dummy* to characterize the baseline performance when we have no correct prior trials for the KC. Indeed, by itself, the *dummy* provides some small improvement since it marks a one-time increase in the prediction after the first correct response is counted. The main purpose, however, is to allow the coefficient for the effect of the prior correct median to be fit freely without the 0 placeholder data values affecting this result.

According to the above correct RT-related component *median$_{corRT}$*, we computed a new feature to capture how the learner's correct RT during the practice process fluctuates around their estimated ideal *fluent$_{RT}$*. The logic behind this feature calculation is that if a learner's correct RT fluctuation for a specific KC is large, even if they just answered the question correctly, the memory traces for the KC maybe still unstable, and the learner probably needs more practice trials on the same KC. The calculation is straightforward, for each learner i, for each trial t, the *fluent$_{RT}$* is subtracted from the *median$_{corRT}$*. The new feature is labeled as *fmedian$_{corRT}$*.

2.2.3 Logistic Knowledge Tracing (LKT) package in R

For logistic regression models, like PFA, the additive nature of features increases their flexibility, making it easy for researchers to add new or drop out old features and build their models. We used the *'LKT'* package [30], which makes the logistic model-building and parameter-searching processes simpler by reducing high-level technical skills and knowledge demands for researchers. For example, the models in this paper were run with single calls to LKT following the data preparation for latency analysis. The LKT code has been publicly shared as an R package in GitHub, and examples with detailed notes are available for reference [31].

2.3 Datasets and Data Preprocessing

The model comparison was conducted across several datasets to examine the improvement from the addition of the correct RT-related features we mentioned above. For calculating the *median$_{corRT}$* and the estimation of *fluent$_{RT}$*, the dataset needs to include a column identifying the time elapsed between the start of the presentation of the specific practice trial and the response reaction made by the individual learner. We used the time from the first seeing the question to the learner's first action as our RT measurement by assuming that this time duration reflected the learner's retrieval time. Specifically, for multiple-choice questions, the learners' response was measured by the mouse click; for short-answer and cloze questions, the response duration was from the first keypress. Furthermore, for fitting logistic models in LKT, columns are required to identify the learners' deidentified id, response accuracy (correct or incorrect), KC id, and the practice opportunity of each KC for the individual learner. We expected that the model predictiveness improvement after incorporating RT-related features should generalize across datasets with different learning materials and formats of practice trials.

The same data preprocessing criteria were applied to all datasets by adopting the procedure of Pavlik and colleagues [30]. Within each dataset, students with less than 25 observations were omitted. KCs with less than 300 observations overall were also omitted. Extreme correct RT outliers (>95th percentile) were winsorized to equal the 95th percentile correct RT values. Missing RT values were imputed with the overall median trial duration for the student. Observations relevant to instructions, learning and review trials, or hints were omitted since we focused on RT values from learners' practice attempts for this study. Furthermore, learners whose accuracy values during the practice session were less than the probability of a random guess were omitted (less than 25%). We used 25% as a general accuracy criterion to maintain consistency across all datasets.

2.3.1 Dataset1. Chinese Vocabulary Pronunciation Memory Multiple-Choice Questions

Dataset 1 was from an experiment designed to explore the best practice context and review spacing schedule for learners to remember the pronunciation of foreign vocabulary words. The learning materials were 27 aural Chinese words. The experiment was conducted by using an online Flashcard learning system. Participants were recruited from Amazon's Mechanical Turk. The format of practice trials was multiple-choice. For each trial, learners were asked to select the correct meaning of the aural Chinese word they had just heard. Learners have 5 seconds to make their choice. Correct answers were provided for learners after their incorrect attempts, and they were encouraged to learn from the feedback within 5 seconds. The 5-second response threshold was chosen because for such a simple task it results in very little truncation of the latency distribution and prevents outlier data from being collected, preferring to mark such unlikely long-duration responses wrong [29]. One Chinese word pronunciation was seen as a unique KC. After data cleaning, 190 learners and a total of 39,282 observations, of which 23,981 correct observations were retained in dataset 1.

2.3.2 Dataset 2. Japanese-English Word Pairs Short Answer Questions

Dataset 2 was from an experiment in optimal learning [9], Experiment 2. The experiment was designed to investigate the effectiveness of an optimal difficulty threshold adaptive scheduling for improving learners' memory retention. The learning materials were 30 Japanese-English word pairs. Participants were recruited from Amazon's Mechanical Turk. All practice trials were short-answer questions, and learners were asked to type in English translations after seeing Japanese words. One unique Japanese-English word pair was seen as a unique KC. The initial dataset included 72,455 observations from 291 adult learners, after data cleaning, 262 learners and a total of 59,885 observations were retained in the dataset, of which 42,482 correct observations were retained.

2.3.3 Dataset 3. Statistics Content Cloze Questions

Dataset 3 from practice with cloze sentences about introductory statistics was downloaded from the Memphis Datashop repository (https://datashop.memphis.edu) [16]. The experiment was designed to explore the effect of spacing schedules and repetition of KCs on learners' memory of simple statistical concepts. The learning materials were 36 sentences about different statistical concepts. Participants were recruited from Amazon's Mechanical Turk. All practice trials were cloze items, and learners were asked to type in the missing word for each sentence. The initial dataset consisted of 58,316 observations from 478 learners. After data screening, 462 learners and a total of 53,277 observations were retained, of which 29,708 were correct observations.

3. RESULTS

3.1 The *fluent$_{RT}$* Estimation Results

Within each dataset, we used Equation 3 and the *optim* function from the *'stats'* R package [35] to estimate the ideal *fluent$_{RT}$* value for each learner. We also calculated the correlation between the learners' estimated *fluent$_{RT}$* and their average RT during the practice session (*average$_{RT}$*). Table 2 shows the descriptive statistics and correlation test results for all three datasets.

Table 2. Descriptive statistics for estimated *fluent$_{RT}$* and its correlation with *average$_{RT}$*

Dataset	***Fluent$_{RT}$* M (SD)**	***Average$_{RT}$* M (SD)**	***Fluent$_{RT}$* and *Average$_{RT}$* correlation**
1	1381.717 (558.811)	1917.562 (655.858)	0.898***
2	2442.846 (855.596)	3231.382 (1132.548)	0.934***
3	3883.096 (956.854)	5910.799 (1215.972)	0.908***

Note. *** p <.001

First, we found a highly positive correlation between the learner's estimated neural integration time (*fluent$_{RT}$*) and motor response time A_i, and the learner's *average$_{RT}$* during the practice session in all datasets. The consistent highly positive correlation suggested that learners' *average$_{RT}$* reflected their neural integration time and motor response time which is reasonable since the individual differences in neurons' response speed. Second, individual differences in neural integration time and motor response time were observed from the *fluent$_{RT}$*. For instance, the estimated *fluent$_{RT}$* of two learners with different response speed tendencies from Dataset 1(the multiple-choice dataset) was shown in Figure 1. It was clear that learner A tended to respond faster than learner B. Based on each learner's correct RT history, the estimated neural integration time and motor response time for learner A was only 759.56 milliseconds, while for learner B, 1830.72 milliseconds corresponded to fluent responding.

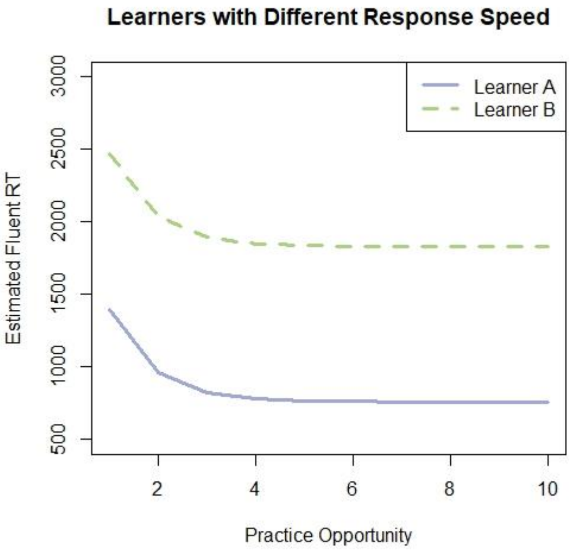


Figure 1. Estimated *fluent$_{RT}$* as a function of the practice opportunity for two learners with different response speed from Dataset 1 (Multiple-Choice Dataset)

3.2 Model Fit and Comparison Results

Five models were fitted to the three datasets. Table 3 shows the features included in each model. The $ operator produces a unique coefficient for each learner and each KC. For example, the 'intercept$learner' feature fits a unique intercept for each learner. While for features without the $ operator, a single coefficient would be fit for the feature. All features shown in Table 3 represent independent variables in logistic regression. The third model (PFA$_{dummy}$) we built here was used as a baseline model to split the unique effects of RT-related features, *median$_{corRT}$* and *fmedian$_{corRT}$*, which we were most interested in.

Table 3. Features included in each model

Model	Features
1_PFA	intercept$learner + intercept$KC + linesucKC + linefailKC
2_PFA$_{fluentRT}$	intercept$learner + intercept$KC + linesucKC + linefailKC + fluent$_{RT}$$learner
3_PFA$_{dummy}$	intercept$learner + intercept$KC + linesucKC + linefailKC + dummy
4_PFA$_{mediancorRT}$	intercept$learner + intercept$KC + linesucKC + linefailKC + dummy + median$_{RT}$$learner
5_PFA$_{fmediancorRT}$	intercept$learner + intercept$KC + linesucKC + linefailKC + dummy + fmedian$_{RT}$$learner

Table 4 shows the model comparison and five-fold unstratified cross-validation results. According to McFadden's R^2 and Akaike information criterion (AIC) values, we can examine whether the predictiveness of standard PFA is improved after incorporating RT-related features. By inspecting the averaged R^2 after 5-fold cross-validation, we want to ensure that the improvement is not caused by over-fitting.

Table 4. Model comparison and cross-validation results

	Model Comparision		
Model	**R^2 (AIC)**	**Δ R^2 (Δ AIC)**	**CV R^2**
Multiple-Choice Dataset			
PFA	0.1082 (47273.57)	-	0.0984
PFA$_{fluentRT}$	0.1082 (47275.56)	-	0.0984
PFA$_{dummy}$	0.1315 (46053.14)	0.0233 (-1220.43)	0.1215
PFA$_{mediancorRT}$	0.1417 (45516.02)	0.0102 (-537.11)	0.1318
PFA$_{fmediancorRT}$	0.1438 (45409.95)	0.0123 (-643.18)	0.1338
Short-Answer Dataset			
PFA	0.1966 (58704.92)	-	0.1855
PFA$_{fluentRT}$	0.1966 (58706.92)	-	0.1855
PFA$_{dummy}$	0.2133 (57503.39)	0.0166 (-1201.53)	0.2022

	Model Comparision		
Model	**R^2** **(AIC)**	**Δ R^2** **(Δ AIC)**	**CV R^2**
PFA$_{mediancorRT}$	0.2217 (56898.67)	0.0084 (-604.71)	0.2105
PFA$_{fmediancorRT}$	0.2163 (57287.38)	0.0030 (-216.00)	0.2052
Cloze Dataset			
PFA	0.2752 (54229.77)	-	0.2564
PFA$_{fluentRT}$	0.2752 (54232.25)	-	0.2564
PFA$_{dummy}$	0.2920 (53002.99)	0.0168 (-1226.78)	0.2728
PFA$_{mediancorRT}$	0.2929 (52940.5)	0.0008 (-62.49)	0.2736
PFA$_{fmediancorRT}$	0.2923 (52979.33)	0.0003 (-23.65)	0.2731

Note. Δ McFadden's R^2 calculates the difference between PFA$_{dummy}$ and PFA; the difference between PFA$_{mediancorRT}$ and PFA$_{dummy}$; the difference between PFA$_{fmediancorRT}$ and PFA$_{dummy}$, respectively. Values reflect the pure influence predicted by the *median*$_{corRT}$ and *fmedian*$_{corRT}$ features.

First, the model comparison results showed that adding the *fluent*$_{RT}$ feature did not improve the predictiveness of standard PFA much for all three datasets. This suggested that the learner's overall processing speed contributed little to predicting their future performance. Second, after incorporating the *median*$_{corRT}$ and the *fmedian*$_{corRT}$ features to model learning-correlated speedup, the predictiveness of the standard PFA was improved most in the Multiple-choice dataset (Dataset 1). At the same time, the improvement was not crucial for both the Short-answer dataset (Dataset 2) and the Cloze-question dataset (Dataset 3). Third, the *dummy* feature caused stable improvement for the standard PFA model across three datasets, indicating that incorrect trials before the first correct response of the learner, perhaps represented the learner's encoding phase [44].

4. DISCUSSION

When predicting learners' future performance, accuracy-based features have been used in various learner modelings, such as knowledge tracing [9, 12] and logistic regression [7, 8, 30, 32]. Recently, some researchers have argued that learners' response time (RT) during practice is also informative for predicting their future performance [21, 23, 24, 25, 26, 41, 42]. The key theoretical rationale behind such assumptions is the strength theory [27] which emphasizes the positive correlation between the correct RT and the strength of memory traces. Quicker correct responses indicate more stable memory traces have been generated than slower correct responses.

Following the strength assumption, in the present research, we calculated two RT-related features, then investigated how much the predictiveness of the standard performance factor analysis model (PFA) can be improved after combining the learner's RT history. The first feature, *median*$_{corRT}$, captures the sequential median correct RT for the specific KC of an individual learner. The second feature, *fmedian*$_{corRT}$, captures how the learner's median correct RT fluctuates around their estimated ideal fluent RT (*fluent*$_{RT}$). The *fluent*$_{RT}$ for each learner is estimated using the exponential law of practice function [13]. The intercept of the exponential function is seen as the *fluent*$_{RT}$ which represents the neural integration time and motor response time without retrieval time, in other words, the assumption here is that the intercept reflects the minimum RT needed for an individual learner to correctly answer a specific KC after reaching to the automatic level.

Our results show that the improvement of standard PFA by *median*$_{corRT}$ and *fmedian*$_{corRT}$ features on the learner's future performance are constrained by the practice questions format. For multiple-choice questions, the observed accuracy perhaps cannot precisely reflect the learner's latent learning processes since the correct responses might be caused by guessing. Thus, after incorporating RT-related features, such measurement imprecision of accuracy can be somewhat offset, resulting in improvements of predictiveness. While for short-answer and cloze questions, the lack of precision of the latency in representing strength limits the method's effectiveness.

One exciting aspect of the research was the unexpected benefit of using the dummy variable we computed to differentiate trials before the first correct response from trials after a correct response. This improvement is not directly related to reaction time hypotheses we had, and indicates future work is needed to understand this result and its generality (though it was more broadly applicable than the RT terms themselves). We speculate that the *dummy* feature may trace the transition between stages of learning. Perhaps indicating the student is moving from an encoding to responding stage of learning similar to what has been proposed in cognitive theories of skill acquisition [36, 44]. Another possible underlying construct traced by the *dummy* feature may be relevant to the moment-to-moment learning proposed by Baker and colleagues for Bayesian knowledge tracing [6]. For instance, the *dummy* feature which detects the first correct response in a series of responses could indicate a learner's state change between unlearned and learned at a coarse grain size.

Limitations of the present research should be noted here as future research directions. One limitation is the method we used to estimate the learner's ideal *fluent*$_{RT}$. In Equation 3, for simplifying calculations, B and α values were assumed as the same for all learners across all to-be-learned items to keep the parsimonious model. Consequently, the practice curves for different learners have the same shape and are only different in the vertical y-coordinate direction (see Figure 1). We also estimated the same A value for each learner across all items. These simplifications may constrain the implications of RT-related features since the same learner's fluent RT for different items is variable, and more difficult items typically require longer RT than easier items [15]. Thus, in future research, more precise estimated *fluent*$_{RT}$ values for each specific KC may be required before incorporating RT-related features in the real-time adaptive scheduling system. Another limitation in the current research is that our results are most relevant to simple-fact memory tasks. Thus, one further research direction is how to generalize the RT-related features to more complex tasks such as arithmetic. However, different from simple memory tasks, how to accurately decompose learners' RT data to precisely reflect their cognitive processes involved in complex tasks requires more effort before generating the RT-related features.

5. ACKNOWLEDGMENTS

This work was partially supported by the National Science Foundation Learner Data Institute (NSF #1934745).

6. REFERENCES

[1] Anderson, J. R. 1981. Interference: The relationship between response latency and response accuracy. *Journal of Experimental Psychology: Human Learning and Memory*, 7, 5, 326–343. DOI= https://doi.org/10.1037/0278-7393.7.5.326.

[2] Anderson, J. R., & Schooler, L. J. 1991. Reflections of the environment in memory. *Psychological Science*, 2, 6, 396–408. DOI= https://doi.org/10.1111/j.1467-9280.1991.tb00174.x.

[3] Anderson, R. B. 2001. The power law as an emergent property. *Memory & Cognition*, 29, 7, 1061–1068. DOI= https://doi.org/10.3758/bf03195767.

[4] Atkinson, R. C. 1972. Ingredients for a theory of instruction. *American Psychologist*, 27, 10, 921–931. DOI= https://doi.org/10.1037/h0033572.

[5] Baker, R.S.J.d., Corbett, A.T., Aleven, V. 2008. More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian knowledge tracing. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds) *Intelligent Tutoring Systems*. ITS 2008. Lecture Notes in Computer Science, vol 5091. Springer, Berlin, Heidelberg. DOI= https://doi.org/10.1007/978-3-540-69132-7_44.

[6] Baker, R. S., Goldstein, A. O., & Heffernan, N. T. 2011. Detecting learning moment-by-moment. *Artificial Intelligence in Education*, *21*(1), 5–25. DOI= https://doi.org/10.3233/jai-2011-015

[7] Cen, H., Koedinger, K., & Junker, B. 2006. Learning factors analysis: A general method for cognitive model evaluation and improvement. *Intelligent Tutoring Systems*, vol 4053, 164–175. DOI= https://doi.org/10.1007/11774303_17.

[8] Chi, M., Koedinger, K. R., Gordon, G. J., Jordan, P. W., & VanLehn, K. 2011. Instructional factors analysis: A cognitive model for multiple instructional interventions. *Educational Data Mining*, 61–70. DOI= https://doi.org/10.1184/r1/6475808.v1.

[9] Corbett, A. T., & Anderson, J. R. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modelling and User-Adapted Interaction*, 4, 4, 253–278. DOI= https://doi.org/10.1007/bf01099821.

[10] Eglington, L. G., & Pavlik Jr, P. I. 2020. Optimizing practice scheduling requires quantitative tracking of individual item performance. *Npj Science of Learning*, 5, 1. DOI= https://doi.org/10.1038/s41539-020-00074-4.

[11] Eglington, L. G., & Pavlik, Jr, P. I. 2019. Predictiveness of prior failures is improved by incorporating trial duration. *Journal of Educational Data Mining*, 11, 2, 1–19. DOI= https://doi.org/10.5281/zenodo.3554675.

[12] Gervet, T., Koedinger, K. R., Schneider, J., & Mitchell, T. M. 2020. When is deep learning the best approach to knowledge tracing. *Educational Data Mining*, 12, 3, 31–54. DOI= https://doi.org/10.5281/zenodo.4143614.

[13] Heathcote, A., Brown, S., & Mewhort, D. J. K. 2000. The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin & Review*, 7, 2, 185–207. DOI= https://doi.org/10.3758/bf03212979.

[14] Housen, A., & Kuiken, F. 2009. Complexity, accuracy, and fluency in second language acquisition. *Applied Linguistics*, 30, 4, 461–473. DOI= https://doi.org/10.1093/applin/amp048.

[15] Judd, W. A., & Glaser, R. 1969. Response latency as a function of training method, information level, acquisition, and overlearning. *Journal of Educational Psychology*, 60, 4, Pt.2, 1–30. DOI= https://doi.org/10.1037/h0020058.

[16] Koedinger, K. R., Baker, R.S.J.d., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J. 2010. A Data Repository for the EDM community: The PSLC DataShop. In *Handbook of Educational Data Mining*. Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.J.d. (Eds.). Boca Raton, FL: CRC Press.

[17] Koedinger, K. R., Corbett, A. T., & Perfetti, C. 2012. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 36, 5, 757–798. DOI= https://doi.org/10.1111/j.1551-6709.2012.01245.x.

[18] Krasne, S., Stevens, C. D., Kellman, P. J., & Niemann, J. T. 2020. Mastering electrocardiogram interpretation skills through a perceptual and adaptive learning module. *AEM Education and Training*, 5, 2. DOI= https://doi.org/10.1002/aet2.10454.

[19] Lindsey, R. 2014. Probabilistic models of student learning and forgetting (Doctoral dissertation, University of Colorado at Boulder).

[20] Lindsey, R., Mozer, M. C., Cepeda, N. J., & Pashler, H. 2009. Optimizing memory retention with cognitive models. In *9th International Conference on Cognitive Modeling (ICCM)*, A. Howes, D. Peebles, & R. Cooper, Eds. Manchester, UK: ICCM, 74–79.

[21] Lindsey, R. V., Lewis, O., Pashler, H., & Mozer, M. C. 2010. Predicting students' retention of facts from feedback during training. In *Proceedings of the 32nd annual conference of the cognitive science society* S. Ohlsson & R. Catrambone, Eds. Austin, TX: Cognitive Science Society, 2332–2337.

[22] Lindsey, R. V., Shroyer, J. D., Pashler, H., & Mozer, M. C. 2014. Improving students' long-term knowledge retention through personalized review. *Psychological Science*, 25, 3, 639–647. DOI= https://doi.org/10.1177/0956797613504302.

[23] Mettler, E., & Kellman, P. J. 2014. Adaptive response-time-based category sequencing in perceptual learning. *Vision Research*, 99, 111–123. DOI= https://doi.org/10.1016/j.visres.2013.12.009.

[24] Mettler, E., Massey, C. M., & Kellman, P. J. 2016. A comparison of adaptive and fixed schedules of practice. *Journal of Experimental Psychology: General*, 145, 7, 897–917. DOI= https://doi.org/10.1037/xge0000170.

[25] Mettler, E., Massey, C. M., & Kellman, P. J. 2010. Improving adaptive learning technology through the use of response times. *Cognitive Science*, 33, 33. DOI= https://escholarship.org/content/qt2xs4n8wz/qt2xs4n8wz.pdf?t=op2jwo.

[26] Mozer, M. C., & Lindsey, R. V. 2016. Predicting and improving memory retention: Psychological theory matters in the big data era. *Big Data in Cognitive Science*, 43–73. DOI= https://doi.org/10.4324/9781315413570-8.

[27] Murdock, B. B. 1985. An analysis of the strength-latency relationship. *Memory & Cognition*, 13, 6, 511–521. DOI= https://doi.org/10.3758/bf03198322.

[28] Newell, A. & Rosenbloom, P. S. 1993. Mechanisms of skill acquisition and the law of practice. In *The Soar Papers (Vol. 1): Research on Integrated Intelligence,* 81–135 (MIT Press).

[29] Pavlik, P. I. 2007. Understanding and applying the dynamics of test practice and study practice. *Instructional Science, 35*(5), 407–441.DOI= https://doi.org/10.1007/s11251-006-9013-2

[30] Pavlik Jr, P., Eglington, L., & Harrell-Williams, L. 2021. Logistic knowledge tracing: A constrained framework for learner modeling. *IEEE Transactions on Learning Technologies*, 14, 5, 624–639. DOI= https://doi.org/10.1109/tlt.2021.3128569.

[31] Pavlik Jr, P., Eglington, L. "LKT." github.com. https://github.com/Optimal-Learning-Lab/LKT (accessed Dec. 1, 2020).

[32] Pavlik Jr, P., Cen, H., & Koedinger, K. R. 2009. Performance factors analysis: A new alternative to knowledge tracing. *Artificial Intelligence in Education*, 1, 531–538. DOI= https://doi.org/10.3233/978-1-60750-028-5-531.

[33] Pavlik Jr, P. I., & Anderson, J. R. 2008. Using a model to compute the optimal schedule of practice. *Journal of Experimental Psychology: Applied*, 14, 2, 101–117. DOI= https://doi.org/10.1037/1076-898x.14.2.101.

[34] Rasch, G. 1966. An individualistic approach to item analysis. In *Readings in mathematical social science*, Lazarsfeld PF, Henry NW, editors. Chicago: Science Research Associates; 89-108.

[35] R Core Team. 2020. *R: A language and environment for statistical computing* [Computer software]. R Foundation for Statistical Computing. http://www.R-project.org/

[36] Rickard, T. C. 1997. Bending the power law: A CMPL theory of strategy shifts and the automatization of cognitive skills. *Journal of Experimental Psychology: General*, 126, 3, 288–311. DOI= https://doi.org/10.1037/0096-3445.126.3.288.

[37] Ritter, S., Yudelson, M., Fancsali, S. E., & Berman, S. R. 2016. How mastery learning works at scale. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale,* 71-79.

[38] Schmucker, R., Wang, J., Hu, S., & Mitchell, T. 2022. Assessing the performance of online students - new data, new approaches, improved accuracy. *Journal of Educational Data Mining*, 14, 1, 1–45. DOI= https://doi.org/10.5281/zenodo.6450190.

[39] Sense, F., van der Velde, M., & van Rijn, H. 2021. Predicting university students' exam performance using a model-based adaptive fact-learning system. *Journal of Learning Analytics*, 8, 3, 155–169. DOI= https://doi.org/10.18608/jla.2021.6590.

[40] Sense, F., Meijer, R. R., & van Rijn, H. 2018. Exploration of the rate of forgetting as a domain-specific individual differences measure. *Frontiers in Education*, 3. DOI= https://doi.org/10.3389/feduc.2018.00112.

[41] Sense, F., Behrens, F., Meijer, R. R., & van Rijn, H. 2016. An individual's rate of forgetting is stable over time but differs across materials. *Topics in Cognitive Science*, 8, 1, 305–321. DOI= https://doi.org/10.1111/tops.12183.

[42] Sense, F., & van Rijn, H. 2022. Optimizing fact-learning with a response-latency-based adaptive system. DOI= https://doi.org/10.31234/osf.io/chpgv.

[43] Shih, B., Koedinger, K., & Scheines, R. 2010. A response time model for bottom-out hints as worked examples. In *Proceedings of the 1st International Conference on Educational Data Mining*, R. S. Baker and J. E. Beck Eds., Montreal, Canada, 117-126.

[44] Tenison, C., Fincham, J. M., & Anderson, J. R. 2016. Phases of learning: How skill acquisition impacts cognitive processing. *Cognitive Psychology*, 87, 1–28. DOI= https://doi.org/10.1016/j.cogpsych.2016.03.001.

The Right To Be Forgotten and Educational Data Mining: Challenges and Paths Forward

Stephen Hutt
University of Denver
stephen.hutt@du.edu

Sanchari Das
University of Denver
sanchari.das@du.edu

Ryan S. Baker
University of Pennsylvania
rybaker@upenn.edu

ABSTRACT

The General Data Protection Regulation (GDPR) in the European Union contains directions on how user data may be collected, stored, and when it must be deleted. As similar legislation is developed around the globe, there is the potential for repercussions across multiple fields of research, including educational data mining (EDM). Over the past two decades, the EDM community has taken consistent steps to protect learner privacy within our research, whilst pursuing goals that will benefit their learning. However, recent privacy legislation may cause our practices to need to change. The right to be forgotten states that users have the right to request that all their data (including deidentified data generated by them) be removed. In this paper, we discuss the potential challenges of this legislation for EDM research, including impacts on Open Science practices, data modeling, and data sharing. We also consider changes to EDM best practices that may aid compliance with this new legislation.

Keywords

Data Mining, User Modeling, Right to Be Forgotten, Data Privacy, GDPR

1. INTRODUCTION

Data from learners is a critical component of Educational Data Mining (EDM). This data can include demographic information, performance data, and interactions with educational resources such as games, intelligent tutoring systems, and online learning platforms. This data is essential for core goals within EDM research, including contributing to learning theory [5], informing learning interventions [48], creating dynamic and personalized learning technology [30], and informing education policy [3]. The collection and use of learner data raises a number of ethical and legal concerns, including privacy and data security. However, with proper safeguards in place, such data can have significant benefits for both students and educators. By providing valuable insights into student learning, EDM can support the development of more effective educational practices and policies, and ultimately improve student outcomes.

The data that facilitates EDM research can often include personal identifying information (PII) and other protected information. As such, there has been increased attention to privacy protection in recent years. De-identification (removing or obscuring PII from data) has become a standard practice for data sharing. Similarly, researchers have used secure platforms to store and share data that leverage access controls, encryption, and other security measures to safeguard the data. Furthermore, there are also research methods such as Differential Privacy [13, 19], which aims to provide privacy-preserving data analysis by adding noise to the data to mask any information about individuals while preserving the overall trends and patterns. There has been considerable research attention to finding the balance between data privacy and having the data required to drive meaningful insights [32, 51] and creating environments where data can be analyzed in its entirety, without being directly shared [29].

Outside of the EDM community, data privacy concerns are also rising. School districts and public advocates have expressed concerns about the increasing amount of education data becoming available at scale (either for commercial or research use) [42, 58]. Klose et al. [34] note that educational repositories have the potential to contribute to identity theft if hacked, and have shared potential solutions to facilitate the storing of educational data. The Student Data Privacy Consortium, meanwhile, has created a template data agreement between educators and researchers. This template requires that any sharing of a dataset (including deidentified data) must be agreed upon by the local education authority on each occasion [57, 59]. Such measures will undoubtedly protect learners, but are onerous to the point that they will likely limit how much data is actually shared, subsequently limiting the potential for research to benefit students.

More broadly speaking, legislators are also considering the issue of user data and are passing laws that govern how it can be collected, used, and shared. In the United States, the Family Educational Rights and Privacy Act (FERPA) has governed many aspects of educational data since 1974, however, it is more general data privacy laws that may have the most impact on research today. The General Data Protection Regulation (GDPR) in the European Union and the Children's Online Privacy Protection Act (COPPA) in the United States each try to protect users and give them more

S. Hutt, S. Das, and R. Baker. The right to be forgotten and educational data mining: Challenges and paths forward. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 251–259, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115655

control over their data when interacting online. Local governments have also taken steps to legislate around how data might be used, for example, the Colorado Privacy Act and the California Consumer Privacy Act (CCPA) also both contain guidelines on user data, both with regard to storage and sharing. With this trend of increased legal guidance around user data, we must consider how legislation might impact research practice, and how to adjust our research practices accordingly.

One such aspect of legislation that may impact EDM research, and the focus of this paper, is the right to erasure - more commonly called the right to be forgotten. This right, included in GDPR, with variations in other legislation, states that a user may request that their data be removed. Given the high volume of learner data that is central to EDM research, this has the potential to impact our research practices. For example, Such removal of data could impact if a scientific result replicates, or create a ripple effect with those with whom the data has been shared. In the remainder of this paper, we present some of the primary challenges the right to be forgotten may impose upon EDM research so that we may be proactive in addressing and understanding the implications of these laws.

2. BACKGROUND

2.1 Privacy Legislation and the Right to be Forgotten

On May 25, 2018, the European Union (EU) implemented the General Data Protection Regulation (GDPR), a comprehensive data protection law. It seeks to unify data protection laws across the EU and replaced the 1995 EU Data Protection Directive. In addition, it gives EU citizens more control over their personal data [62]. Regardless of whether an organization is based in the EU, it must comply with the GDPR if it processes the personal data of EU citizens [16]. The right to erasure, also known as the "right to be forgotten," (RTBF) is one of the GDPR's most significant provisions, relative to previous legislation. When specific requirements are met, EU citizens have the right to request that their personal data be erased under the RTBF [53, 64]. This may occur when a person withdraws their consent to processing their data, for instance, or when the personal data is no longer required for the purpose for which it was collected.

This legislation gives users more control over how their personal information is shared and formalizes an issue that had already been discussed in the courts. In the 2014 case of Google Spain vs. Agencia Espanola de Protección de Datos (AEPD) and Mario Costeja González, the European Court of Justice determined that a person has the right to ask for the removal of links to personal information from a search engine if the information is unreliable, insufficient, irrelevant, or excessive for the data processing. Furthermore, the court ruled that the data controller (in this case, Google Spain) was required to take reasonable steps to notify third-party controllers (any other organization with which the data was shared) of the individual's request. Due to this decision, Google established a procedure for people to submit RTBF requests known as the "right to be forgotten" form [18]. However, the ruling was not absolute. It could be superseded by other rights and interests, such as the right to information and freedom of expression. With the passing of GDPR, there are still elements of ambiguity of supersedence, for example, if the processing of the data is required to carry out a task in the public interest or the exercise of official authority vested in the controller [8, 36].

Similarly worded legislation has been enacted outside the EU, including in the US, Canada, and Asia. For example, the California Consumer Privacy Act (CCPA) was enacted in the US on January 1, 2020. It grants residents of California certain rights regarding their personal data, including the right to ask for the deletion of personal data held by a company (though with less severe penalties than GDPR) [26, 1]. The Personal Information Protection and Electronic Documents Act (PIPEDA) in Canada governs the private sector's gathering, use, and disclosure of personal information. It does not explicitly address the right to be forgotten. However, according to the Office of the Privacy Commissioner of Canada in 2019, the Act grants individuals the right to access and amend their personal information and the freedom to revoke their consent to its collection, use, or disclosure [50, 37].

Comparably, Singapore's Personal Data Protection Act 2012 (PDPA) governs how businesses collect, use, and disclose individuals' personal information [63, 11]. It grants people the right to withdraw their consent for the collection, use, or disclosure of personal information and access and correct their personal data. Additionally, organizations must delete personal data under section 26 of the act when it is no longer required for the purpose for which it was collected. The common theme across each of these laws is that they provide people more control over their data and the ability to ask for the deletion of data that is no longer relevant or necessary. The GDPR has established a high bar for data protection in the EU. However, with varying levels of legislation across the world, remaining in compliance with each of the varying laws can be challenging (especially if a dataset contains users from multiple locations). This can be especially challenging for researchers striving to share data and provide transparency regarding their scientific methods.

2.2 Replicability Crisis

Replication (in this context) refers to the verification of a scientific study's finding through reproduction, either from the same data, or new data following the same design. The purpose of this process is to better understand the reliability, validity, and merit of a study's findings [15]. A study is deemed reproducible if a research team is able to obtain its original results through the execution of its original method on the original or a comparable dataset [22]. "Reproducibility is a minimum necessary condition for a finding to be believable and informative" [6].

Despite this importance, replication studies remain somewhat rare in education research and in data research more broadly. In a study conducted on 400 previously published works from leading artificial intelligence venues, none of the papers analyzed reported all details necessary to fully replicate their work [24]. In a study conducted on 30 published works on text mining, for example, only one of the studies provided source code to replicate their findings [46]. The report cited lack of access to data, computation capacity, and implementation methods as primary barriers to replication.

The lack of replication leads to a surprisingly large proportion of spurious results being widely reported, as reported by the Open Science Collaboration [47]. In their report, the OSC, an open collaboration of scientists that seeks to improve scientific values and practices, replicated a hundred studies from three top psychology journals. Their study found that 64% of the replications conducted failed to obtain statistically significant results. These findings highlight the importance of replication research and the need to validate published findings. As such, a growing body of research has begun advocating for researchers to take more active steps to facilitate replication through Open Science practices.

2.3 Open Science and Open Data

Recent years have seen increasing movements developing in favor of Open Science and Open Data. The Open Science movement involves a variety of initiatives and values aimed at making scientific research more accessible, more transparent, and more reproducible and replicable. Open Science incorporates a number of different elements, including open (public) access to scientific publications, the use of open-source software, and (of particular relevance to this article) Open Data. Though ideas around Open Science have been around for a considerable time [12], the contemporary Open Science movement arguably dates to the Budapest Open Access Initiative [10], which called for open archives and open access journals. So too, scientific data has been shared publicly for a considerable time [43], accelerating with the advent of the public Internet/World Wide Web [25]. However, a large proportion of scientific data remains inaccessible to other scientific researchers [60], much less the general public.

Within education specifically, the amount of data available openly expanded considerably in the first decade of the 21st century, with repositories such as TalkBank [41] and the Pittsburgh Science of Learning Center DataShop [35]. In recent years, many learning platforms have made their data sets public, and the International Educational Data Mining Society has inaugurated a prize for each year's best publicly available data set. Indeed, this year's conference (2023) includes Open science badges to encourage researchers to share data, and materials, and pre-register their analysis. Increasingly, many funding agencies supporting educational research worldwide have begun to require data management plans, with strong encouragement to make data openly available [44, 17], and a recent Executive Order by the U.S. government mandates open access to scientific publications and open sharing of data starting in 2026 [45]. As such, the already increasing moves within the field towards Open Science and Open Data appear likely to expand considerably in the next few years.

3. RIGHT TO BE FORGOTTEN AND EDM

The right to be forgotten (RTBF) can have a significant impact on the practice of researchers in educational data mining. Under RTBF, all data generated by a learner must be removed from databases upon their request. This can be a difficult and time-consuming task, especially if the data is stored in multiple locations, has been shared with colleagues, or even made publicly available. This can result in a ripple effect where the request to remove the data must be passed on to anyone who received a copy of the data, making it difficult to ensure that the request has been completely satisfied. This could lead to researchers or other data providers stopping data sharing altogether, which would considerably slow research progress and disproportionately impact researchers from communities where funding is more scarce.

Moreover, the data covered by RTBF is extensive. Data that has previously been protected, such as personal identifying information (PII), is covered, but so is any additional data generated by that learner. Interaction data generated as a learner plays an educational game, for example, although in many cases not identifying, would still be covered. Similarly, data from intelligent tutoring software, online learning platforms, or MOOCs would all be covered. Thus, a domino effect of data removal occurs, one that, in collaborative systems, may go beyond an individual learner. There are some that argue that such a broad definition of user data is not required under the legislation, and there that there is ambiguity. To our knowledge, the inclusion of data beyond PII has not yet been tested/challenged in the courts, but such a challenge may well happen in the future. It is also worth noting that despite the lack of testing, many organizations (including the authors' universities and other universities) are acting with this broad definition of user data, which may in time set a precedent outside of the courts.

Placing the right to be forgotten into the context of EDM requires complex planning and execution, given that the removal of a learner's data is not as simple as deidentification. Considering the GDPR legislation specifically, data providers would need to remove all data generated by that learner from databases and shared data sets. In order to mitigate the impact of RTBF on EDM research, it becomes necessary for researchers to keep detailed records of who has access to the data and to plan for the possibility of data removal in the future. By necessity, researchers are also required to keep identifiers for all data so that data can be accurately deleted upon request. This means that datasets that would normally be fully deidentified, must now retain some level of identification, potentially creating additional privacy risks. Some mitigation strategies may include using secure data-sharing platforms that allow for selective data removal and data-sharing agreements that include specific provisions for compliance with RTBF legislation. We do not currently know of any published statistics of how many RTBF requests are being made, however, anecdotally, the third author of this paper holds an administrative leadership role involving handling these requests for their university. Although the university is located in the United States, there have been dozens of requests from EU citizens to be removed, with new batches every month. These requests are then legally required to be processed quickly.

There may also be further impacts of RTBF on research practice. For example, what if the data has been published publicly? What if results have been published, and they can no longer be replicated if the analysis were run again? What if the data is in ongoing use? If a current study can not replicate a past finding, should they compare to the published version of the finding or the finding from the current data set? How can we detect scientific fraud when published results can no longer be checked? In the remainder of this section, we consider the potential impacts RTBF may have on the field's practices.

3.1 Data Sharing

The right to be forgotten requires that all learner data be removed. If all of the data is stored in one location, this is a somewhat simple (though potentially time-consuming) task. If the data is stored in multiple locations (e.g., multi-site collaborative projects), the task is more challenging and requires slightly more coordination. Should the data have been shared with colleagues outside the immediate collaboration (for purposes of replication or data sharing), it becomes more challenging still, with perhaps the highest challenge being if the data is shared publicly, with no record of who downloaded it.

The right to be forgotten can create a ripple effect, with the request needing to be passed to anyone who received a copy of the data from the original researcher. This effect could result in a significant amount of time required to remove an individual learner's data. This effort increases drastically if the researchers do not have a clear record of who has the data, and it becomes almost impossible if the data was shared publicly. In this case, a researcher could remove the learner's data from the public posting, but not from everyone who had already downloaded a copy, thus not completing their responsibility.

One option to counter the challenge that the right to be forgotten places upon data sharing, is to simply stop sharing data. To stop publishing datasets online or sharing with colleagues. However, this comes with significant disadvantages. Data collection is expensive [14], if data is not shared, data-driven research (such as data mining) will be limited to those that can afford to collect their own data. This will limit much of the research in our field to data owners (i.e., industry and those able to complete primary data collection), or to data sets from countries with less restrictive regulations. Put simply, should data sharing stop, research progress will be slowed, and this slowdown will have a disproportionate impact on researchers from communities where external funding is more sparse (and therefore it is impractical to collect large data sets). Such an equity issue would take the field backwards, and thus we should consider methods that could facilitate data sharing, without creating this particular ripple effect.

3.2 Replicability

Another major ripple effect of the right to be forgotten is in terms of replicability. As noted above, a disappointingly large proportion of research – even machine learning research, where both the data set and code are both available – is not currently replicable [24, 46]. This lack of replicability has several costs. The first and foremost of these is being able to verify if a prior set of analyses was authentic and correctly conducted.

Unfortunately, the right to be forgotten – under certain interpretations – is likely to considerably worsen this problem, and undo the gains of the last several years. If the data set that a past analysis was run on becomes no longer available, it cannot be replicated. Even the removal of one student from a very large data set presents the possibility that a different model will be obtained, or that goodness metrics or statistical results will shift. The field does not currently have methods tailored to determining how much shift could plausibly be expected if one or more students are omitted, and it will be difficult to develop a general framework for predicting shift of this nature, across the broad range of algorithms and models currently used in educational data mining and data science more generally. The field also does not have practices for what to do if – for example – a published finding is no longer obtained within the reduced version of the data set now available. With the right to be forgotten, building on past research will become more difficult and even identifying scientific fraud will be impaired.

Similarly, the right to be forgotten places requirements on data that is ”no longer required or relevant” [62]. It is difficult to tell when data is no longer required or relevant, if replication is a future possibility. It is not presently clear if storing data for the purposes of replication will be considered ”required” or ”relevant” under the legislation. This, in turn, means that further challenges may appear as the practical implications of the legislation (and its interpretation by the courts) become more clear.

3.3 Progressive Science

In addition to replicating previous work, RTBF can present challenges for *building upon* previous work. There is a chance that RTBF requests will result in the deletion of data that is still useful for research [23] - uses that may not be clear at the time of deletion. Similarly, RTBF may limit our ability to compare new work to previous results [4]. For instance, if we cannot replicate prior work, it becomes impossible to tell if a new algorithm is genuinely an improvement upon past work, particularly if a different validation approach is deemed appropriate. Comparative analysis is a crucial technique for assessing the efficacy of different models and pinpointing potential areas for development and future improvement. For instance, a positive recent trend in research on knowledge tracing is the comparison of models across various data sets [20]. This makes it possible for academics and industry professionals to assess the generalizability of their findings, gauge the robustness of new models, and spot data biases or outliers. However, it might be challenging to make these kinds of comparisons and to assess the efficacy of various models if data is removed in response to RTBF requests – two papers could obtain different findings for the same algorithm and supposed same data set.

3.4 Longitudinal Followup

RTBF may also impact the ability to conduct longitudinal studies and monitor student progress. If students exercise their right to be forgotten, comparing and linking data on future outcomes will become more challenging [21]. The goal of longitudinal followup research is often to determine if a curriculum or pedagogy that was effective in the short-term has longer-term benefits for students, particularly students in historically underrepresented groups who are less well-served by current educational systems [52]. Students in historically underrepresented groups are more likely to opt-out of their data being used [40] – in combination with the RTBF, this means that longitudinal research may only be able to demonstrate long-term effectiveness for students who are already well-served. If an analysis does not explicitly check for consistency of effects across demographic groups, this may lead to an approach being adopted despite (unknown) lower effectiveness for historically underrepresented students.

3.5 Models

One consistent area of EDM research has been the training of statistical and machine-learned models. These models are then integrated into learning environments, dashboards, etc. to provide better learning experiences, and analytics [61]. For example, in [31], models of engagement were trained on data collected from learners, and were later used to create a more adaptive intelligent tutor that responded to student engagement and improved learning [30]. Processes such as these allow the research of the EDM community to directly reach learners and broaden our overall impact.

Currently unclear in legislation is how (and whether) data products are different from the data itself. Consider a machine-learned model from 100 learners' data. That model has embedded in it some representation of the 100 learners. It is likely heavily transformed, and unlikely that the original data could be recreated, but still, the model would be different if only 99 learners' data had been included. The model is a product of the data collected from each of the 100 learners. If a learner enacts their right to be forgotten, must they also be removed from their data's product, the model? Must the model be re-fit?

In large-scale machine learning (such as that conducted by Google), the removal of an individual user likely wouldn't change the model too much. However, given the small Ns often seen in EDM research, the impact could be far greater, and would require increased time on behalf of the research team and place a burden on often limited resources. The difference between data and data product is currently ambiguous in legislation. One interpretation is that an existing model needn't change, but any refinement of the model would need to exclude a learner who had requested to be forgotten. As legislation of this kind becomes more widespread, it is likely that this issue will be considered, and potentially clarified. This clarification may come from legislators, researchers, industry leaders, or the courts. In the meantime, it is important that the EDM community be conscious of this issue, and be involved as data privacy laws are refined. Only by being part of the ongoing discussion surrounding legislation can we ensure that all possible use cases of data are being considered.

4. PATHS FORWARD

The RTBF aims to protect learners and safeguard student privacy, a goal that EDM researchers generally agree with. However, its exact application in EDM has the potential to limit research, and force steps backward in replicability and Open Science Practices. As such, the EDM community should work now to find ways to achieve a balance between research needs (e.g., the need for comparative analysis and data-driven research) and the emerging rights of students to be forgotten.

Given the differences by location, knowing for certain if you are in compliance with privacy legislation can be challenging. We, therefore, advocate for the generation of new best practices in EDM. Such practices could be standardized across the community, and ensure that a researcher is in compliance even with the strictest of RTBF requirements. Striking a balance between research and privacy will not be perfect, but by developing standards as a community, we will have generated a common evaluation point for our research and privacy standards for the learners we work with.

In addressing the challenges described above, we can build on work from colleagues in Healthcare especially [33, 55], where some of these issues have already been tackled. Similarly, we can extend work in our own field that has considered privacy-preserving open science techniques. For example, a recent special issue of the British Journal of Educational Technology reported on technical frameworks for ethical and trustworthy education research [38].

4.1 Privacy-Preserving Live Data Sharing

One possibility for tackling challenges posed to research by RTBF is privacy-preserving Data Sharing. By keeping a live copy of data in a central location, we mitigate a number of the concerns raised in section 3.1. By recording who is authorized to access data, effective logging can be implemented, and downloading or converting can be restricted. An additional benefit of such approaches is they are typically a more accessible way to share data, real-time access to data can be provided without the need for downloading or converting data, which can support those who use assistive technologies.

However, this does not present a perfect solution. Though easier to control the ripple effect of data sharing, implementing the necessary controls to guarantee compliance with these laws may be more challenging. GDPR requires companies handling the data of EU citizens to protect that data, including by implementing privacy by design and by default. Because it can be more challenging to monitor and regulate how data is used in real-time, live data sharing can make it more difficult to comply with these requirements. For instance, it may be challenging to guarantee that authorized users only access data (as opposed to downloading it) or that it is being used for intended purposes. Some of the potential solutions include leveraging cloud services that can satisfy these requirements somewhat easily, as well as the addition of new controls. In such cases, however, a researcher is then relying on a third party to ensure that the solution is compliant. That said, it is not clear if stakeholders (parents, school administrators, etc.) would be in support of the use of third-party data sharing, meaning more exploration of such a solution is needed. Similarly, live data may be susceptible to malicious activity such as hacks - invoking concerns raised in [34]. More research is required to fully comprehend the implications of live data sharing and to determine best practices for overcoming the difficulties presented.

4.2 Privacy Preserving Enclaves

Privacy-preserving enclaves enable the processing and analysis of sensitive data while preserving its integrity and confidentiality [39]. These enclaves isolate a secure environment from the rest of the system using hardware and software based security mechanisms. One such enclave is IntelSGX [54, 56]. To create a secure environment for running code and storing data, Intel SGX combines hardware and software security features. Even if the rest of the system is compromised, this isolation guarantees that data and computations are shielded from unauthorized access or manipulation [9, 49]. The ability to enable privacy-preserving live data sharing is one of the main advantages of privacy-preserving enclaves. Real-time data processing and analysis are constrained by traditional methods for sharing sensitive data, such as differential privacy or encryption. On the other hand, privacy-preserving enclaves allow sensitive data to be processed and analyzed in a secure setting without compromising the privacy of the people linked to the data.

An EDM-specific example of a privacy-preserving enclave is the MOOC Replication Framework (MORF) which offers a secure environment for the replication and analysis of massive open online course (MOOC) data [29]. Millions of students from all over the world now take part in MOOCs, which have grown in popularity in recent years. However, the data produced by these MOOCs is sensitive, in that students may reveal personal details on discussion forums, which are challenging to perfectly redact at scale [7]. MORF presents a framework for analyzing MOOC data and replicating past analyses without compromising student privacy. MORF allows users to submit analysis code (in any programming language). This code is then run on the MORF database, and the results are provided to the user, without ever having direct access to the data itself. MORF relies upon a variety of security methods implemented within Amazon Web Services, as well as software based protocols that control the output provided to a user (e.g., a user cannot submit a script to extract the data)[2].

Due to privacy concerns, data is frequently kept private in MOOC research, making it challenging to confirm and validate the results of earlier research. MORF provides accessibility for researchers without compromising learner privacy. As such, MORF offers a potential blueprint for privacy preserving data sharing in the future.

These approaches are not without challenges, however. Keeping the underlying hardware and software secure can be a significant challenge. Intel SGX depends on the operating system and hardware security for a secure environment to run code and store data [65]. However, many security flaws in Intel SGX have been found, raising questions about the security of these enclaves. These enclaves' performance is another drawback because privacy-preserving techniques often increase the system's computational and latency overhead, making them less suitable for some use cases. As a result, it's crucial to weigh the trade-offs and ensure that the advantages outweigh the drawbacks. In addition, it can be harder for researchers to work on platforms with the restrictions that privacy-protecting enclaves such as MORF enforce, such as the inability to direct view data or to use unrestricted outputs for debugging. It should also be noted that this approach does not directly address issues of replicability, though it does take steps to prevent the ripple effect.

4.3 Engaging with the Legislative Process

As this legislation evolves and the practicalities are considered and ruled upon in the courts, there will likely be calls for participation from lawmakers, funding organizations, and advocacy groups. Academic research is not something typically well represented by lobbyists [27], thus, we must more actively engage in the process ourselves. This may take many forms, including response to data collection requests (e.g., surveys, interviews, etc.) from legislators, and organizations working on these problems (such as the National Science Foundation). Another form of participation is providing feedback during comment periods for proposed legislation. Engaging with the legislative process provides a better chance that the needs of scientific work, as well as those of the Open Science and Open Data protocols we are encouraging, are considered by lawmakers.

4.4 Collaboration with other disciplines

The EDM community is not the only one facing these challenges. As such, there may be much to learn from how other research areas and industries tackle these challenges. For example, there are already protocols for sharing data in healthcare that satisfy the Health Insurance Portability and Accountability Act (HIPPA), and its privacy rule [28, 55]. Many of these protocols would also facilitate the kind of logging required to satisfy RTBF requests. By taking advantage of existing advancements, we reduce the burden on our research community and avoid 'reinventing the wheel'.

The push for Open Science and Open Data has been a prominent movement across multiple scientific disciplines. The conflicts discussed in this paper, along with the need to find a balance of compliance with legal restrictions and scientific integrity, are not unique to EDM. By working with our research colleagues across disciplines, we can reach more standardized solutions, which would, among other benefits, support standardized requirements regarding Open Science and Open Data in publishing venues, etc. Similarly, other disciplines may benefit from EDM advances in this area, such as MORF [29].

5. CONCLUSIONS

The right to be forgotten, and similar legislative changes on how we store and use data, are likely to have a significant impact on Educational Data Mining. Though we have noted some potential paths forward to adapt to this change, there is not one clear solution. We encourage others in the EDM community to consider the challenges outlined, the potential solutions, and to be proactive, rather than reactive, to these changes. Such proactivity may take multiple forms: it could include designing data-sharing infrastructure, responding to requests for feedback on proposed legislation changes, or joining conversations regarding the interaction of data privacy and research outside our community. A number of advances have been made with challenges similar to these in the healthcare community, and there is much we could potentially learn from other research environments. The EDM community has had a significant impact on learners and education, and has a continued potential to do so. As legislature changes, we must protect that potential, whilst still providing learners with all the protection they can, and should, receive. It is thus our argument that we should develop and adopt best practices now, to be ready for these changes as they are implemented.

6. REFERENCES

[1] J. S. Baik. Data privacy against innovation or against discrimination?: The case of the California Consumer Privacy Act (CCPA). *Telematics and Informatics*, 52, 2020.

[2] R. Baker, S. Hutt, M. Mogessie, and H. Valayaputtar. Research using the mooc replication framework and e-trials. In *2022 IEEE Learning with MOOCS (LWMOOCS)*, pages 131–136. IEEE, 2022.

[3] R. S. Baker. The Current Trade-off Between Privacy and Equity in Educational Technology. In G. Brown III and C. Makridis, editors, *The Economics of Equity in K-12 Education: Necessary Programming, Policy, and Systemic Changes to Improve the Economic Life Chances of American Students.* Rowman & Littlefield., Lanham, MD, In Press.

[4] G. Bansal and F. F.-H. Nah. Internet privacy concerns revisited: Oversight from surveillance and right to be forgotten as new dimensions. *Information & Management*, 59(3):103618, 2022.

[5] G. Biswas, R. S. Baker, and L. Paquette. Data Mining Methods for Assessing Self-Regulated Learning. In *Handbook of Self-Regulation of Learning and Performance*, pages 388–403. Taylor & Francis Group, 2017.

[6] K. Bollen, J. T. Cacioppo, R. M. Kaplan, J. A. Krosnick, J. L. Olds, and H. Dean. Social, behavioral, and economic sciences perspectives on robust and reliable science. *Report of the Subcommittee on Replicability in Science Advisory Committee to the National Science Foundation Directorate for Social, Behavioral, and Economic Sciences*, 1, 2015.

[7] N. Bosch, R. Crues, N. Shaik, and L. Paquette. ”Hello,[REDACTED]”: Protecting student privacy in analyses of online discussion forums. In *International Conference on Educational Data Mining*. ERIC, 2020.

[8] M. Burri and R. Schär. The reform of the eu data protection framework: outlining key changes and assessing their fitness for a data-driven economy. *Journal of Information Policy*, 6(1):479–511, 2016.

[9] S. Chakrabarti, T. Knauth, D. Kuvaiskii, M. Steiner, and M. Vij. Trusted execution environment with Intel SGX. In *Responsible Genomic Data Sharing*, pages 161–190. Elsevier, 2020.

[10] L. Chan, D. Cuplinskas, M. Eisen, F. Friend, Y. Genova, J.-C. Guédon, M. Hagemann, S. Harnad, R. Johnson, M. L. Kupryte, Rima, I. Rév, M. Segbert, S. de Souza, P. Suber, and J. Velterop. Budapest Open Access Initiative. `https://www.budapestopenaccessinitiative.org/`.

[11] S. Chesterman. After privacy: The rise of facebook, the fall of wikileaks, and singapore's personal data protection act 2012. *Sing. J. Legal Stud.*, page 391, 2012.

[12] D. E. Chubin. Open science and closed science: Tradeoffs in a democracy. *Science, Technology, & Human Values*, 10(2):73–80, 1985.

[13] F. K. Dankar and K. El Emam. Practicing differential privacy in health care: A review. *Trans. Data Privacy*, 6(1):35–67, apr 2013.

[14] G. J. Duncan, N. J. Kirkendall, C. F. Citro, N. R. Council, et al. Data collection costs. In *The National Children's Study 2014: An Assessment*. National Academies Press (US), 2014.

[15] F. Echtler and M. Häußler. Open source, open science, and the replication crisis in HCI. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–8, 2018.

[16] EuropeanCommission. 2018 reform of eu data protection rules. `https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf`.

[17] EuropeanCommission. Horizon Europe (HORIZON) Program Guide., 2022.

[18] E. Frantziou. Further developments in the right to be forgotten: The european court of justice's judgment in case c-131/12, google spain, sl, google inc v agencia espanola de proteccion de datos. *Hum. Rts. L. Rev.*, 14:761, 2014.

[19] A. Friedman and A. Schuster. Data mining with differential privacy. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 493–502, 2010.

[20] T. Gervet, K. Koedinger, J. Schneider, T. Mitchell, et al. When is deep learning the best approach to knowledge tracing? *Journal of Educational Data Mining*, 12(3):31–54, 2020.

[21] A. Goldsteen, G. Ezov, R. Shmelkin, M. Moffie, and A. Farkash. Data minimization for gdpr compliance in machine learning models. *AI and Ethics*, 2(3):477–491, 2022.

[22] S. N. Goodman, D. Fanelli, and J. P. Ioannidis. What does research reproducibility mean? *Science translational medicine*, 8(341):341ps12–341ps12, 2016.

[23] E. Gratton and J. Polonetsky. Droit a l'oubli: Canadian perspective on the global right to be forgotten debate. *Colo. Tech. LJ*, 15:337, 2016.

[24] O. E. Gundersen and S. Kjensmo. State of the art: Reproducibility in artificial intelligence. In *AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[25] S. S. Hale, M. M. Hughes, J. F. Paul, R. S. McAskill, S. A. Rego, D. R. Bender, N. J. Dodge, T. L. Richter, and J. L. Copeland. Managing scientific data: the emap approach. *Environmental Monitoring and Assessment*, 51(1):429–440, 1998.

[26] E. L. Harding, J. J. Vanto, R. Clark, L. Hannah Ji, and S. C. Ainsworth. Understanding the scope and impact of the california consumer privacy act of 2018. *Journal of Data Protection & Privacy*, 2(3):234–253, 2019.

[27] P. Harris and C. McGrath. Political marketing and lobbying: A neglected perspective and research agenda. *Journal of Political Marketing*, 11(1-2):75–94, 2012.

[28] T. Hulsen. Sharing is caring—data sharing initiatives in healthcare. *International Journal of Environmental Research and Public Health*, 17(9), 2020.

[29] S. Hutt, R. S. Baker, M. M. Ashenafi, J. M. Andres-Bray, and C. Brooks. Controlled outputs, full data: A privacy-protecting infrastructure for MOOC data. *British Journal of Educational Technology*, 53(4):756–775, 2022.

[30] S. Hutt, K. Krasich, J. R. Brockmole, and

S. K. D'Mello. Breaking out of the lab: Mitigating mind wandering with gaze-based attention-aware technology in classrooms. In *CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.

[31] S. Hutt, C. Mills, S. White, P. J. Donnelly, and S. K. D'Mello. The Eyes Have It: Gaze-based Detection of Mind Wandering during Learning with an Intelligent Tutoring System. In T. Barnes, M. Chi, and M. Feng, editors, *The 9th International Conference on Educational Data Mining. International Educational Data Mining Society.*, pages 86–93, 2016.

[32] M. Ivanova, G. Grosseck, and C. Holotescu. Researching data privacy models in elearning. In *2015 International Conference on Information Technology Based Higher Education and Training (ITHET)*, pages 1–6, 2015.

[33] M. Jayabalan and M. E. Rana. Anonymizing healthcare records: A study of privacy preserving data publishing techniques. *Advanced Science Letters*, 24(3):1694–1697, 2018.

[34] M. Klose, V. Desai, Y. Song, and E. Gehringer. Edm and privacy: Ethics and legalities of data collection, usage, and storage. In *International Conference on Educational Data Mining*. ERIC, 2020.

[35] K. R. Koedinger, R. S. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A data repository for the edm community: The PSLC DataShop. *Handbook of Educational Data Mining*, 43:43–56, 2010.

[36] C. Kuner. The european commission's proposed data protection regulation: A copernican revolution in european data protection law. *Bloomberg BNA Privacy and Security Law Report (2012) February*, 6(2012):1–15, 2012.

[37] D. Lackey and N. Beaton. The current state of data protection and privacy compliance in canada and the usa. *Applied Marketing Analytics*, 4(4):355–359, 2019.

[38] D. Ladjal, S. Joksimović, T. Rakotoarivelo, and C. Zhan. Technological frameworks on ethical and trustworthy learning analytics. *British Journal of Educational Technology*, 53(4):733–736, 2022.

[39] T. Lee, Z. Lin, S. Pushp, C. Li, Y. Liu, Y. Lee, F. Xu, C. Xu, L. Zhang, and J. Song. Occlumency: Privacy-preserving remote deep-learning inference using sgx. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–17, 2019.

[40] W. Li, K. Sun, F. Schaub, and C. Brooks. Disparities in students' propensity to consent to learning analytics. *International Journal of Artificial Intelligence in Education*, 32(3):564–608, 2022.

[41] B. MacWhinney, S. Bird, C. Cieri, and C. Martell. Talkbank: Building an open unified multimodal database of communicative interaction. In *The Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA).

[42] National Association of Secondary School Principals (NASSP). Student Data Privacy, Feb. 2017.

[43] National Research Council et al. Committee on scientific accomplishments of earth observations from space. 2008. earth observations from space: The first 50 years of scientific achievements.

[44] National Science Foundation. Chapter XI.D.4. In *Proposal & Award Policies & Procedures Guide (PAPPG)*. NSF, 2021.

[45] A. Nelson et al. Memorandum for the heads of executive departments and agencies: Ensuring free, immediate, and equitable access to federally funded research. *Repository and Open Science Access Portal (ROSA)*, 2022.

[46] B. K. Olorisade, P. Brereton, and P. András. Reproducibility in machine learning-based studies: An example of text mining. In *ICML 2017 RML Workshop Reproducibility in Machine Learning*, 2017.

[47] Open Science Collaboration. Estimating the reproducibility of psychological science. *Science*, 349(6251):aac4716, 2015.

[48] V. Owen, M. H. Roy, K. P. Thai, V. Burnett, D. Jacobs, E. Keylor, and R. S. Baker. Detecting wheel-spinning and productive persistence in educational games. In *EDM 2019 - The 12th International Conference on Educational Data Mining*, 2019.

[49] R. Pires, M. Pasin, P. Felber, and C. Fetzer. Secure content-based routing using intel software guard extensions. In *The 17th International Middleware Conference*, pages 1–10, 2016.

[50] J. Pushka. The applicability of the personal information protection and electronic documents act to de-indexing internet search engine results. *Asper Rev. Int'l Bus. & Trade L.*, 19:175, 2019.

[51] J. R. Reidenberg and F. Schaub. Achieving big data privacy in education. *Theory and Research in Education*, 16(3):263–279, 2018.

[52] T. R. Sass, R. W. Zimmer, B. P. Gill, and T. K. Booker. Charter high schools' effects on long-term attainment and earnings. *Journal of Policy Analysis and Management*, 35(3):683–706, 2016.

[53] B. Sealey et al. Has the 2016 general data protection regulation really given consumers more control over their personal data? *LJMU Student Law Journal*, 1:17–41, 2020.

[54] J. Seo, B. Lee, S. M. Kim, M.-W. Shih, I. Shin, D. Han, and T. Kim. Sgx-shield: Enabling address space layout randomization for sgx programs. In *NDSS*, 2017.

[55] B. Shen, J. Guo, and Y. Yang. Medchain: Efficient healthcare data sharing via blockchain. *Applied Sciences*, 9(6), 2019.

[56] M.-W. Shih, S. Lee, T. Kim, and M. Peinado. T-sgx: Eradicating controlled-channel attacks against enclave programs. In *NDSS*, 2017.

[57] Student Data Privacy Consortium. Standard Student Data Privacy Agreement (NDPA Standard Version 1.0) Version 1r7. Technical report, Access of Learning Community, 2021.

[58] Student Privacy Compass. Student Privacy Primer. `https://studentprivacycompass.org/resource/student-privacy-primer/`, 2021.

[59] F. Tazi, S. Shrestha, D. Norton, K. Walsh, and S. Das.

Parents, educators, & caregivers cybersecurity & privacy concerns for remote learning during covid-19. In *CHI Greece 2021: 1st International Conference of the ACM Greek SIGCHI Chapter*, pages 1–5, 2021.

[60] L. Tedersoo, R. Küngas, E. Oras, K. Köster, H. Eenmaa, Ä. Leijen, M. Pedaste, M. Raju, A. Astapova, H. Lukner, et al. Data sharing practices and data availability upon request differ across scientific disciplines. *Scientific Data*, 8(1):1–11, 2021.

[61] K. Verbert, E. Duval, J. Klerkx, S. Govaerts, and J. L. Santos. Learning analytics dashboard applications. *American Behavioral Scientist*, 57(10):1500–1509, 2013.

[62] P. Voigt and A. v. d. Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide.* Springer Publishing Company, Incorporated, 1st edition, 2017.

[63] B. Wong YongQuan. Data privacy law in singapore: The personal data protection act 2012. *International Data Privacy Law*, 2017.

[64] R. Zaman and M. Hassani. Process mining meets gdpr compliance: the right to be forgotten as a use case. In *International Conference on Process Mining - Doctoral Consortium, ICPM-DC*, 2019.

[65] C. Zhao, D. Saifuding, H. Tian, Y. Zhang, and C. Xing. On the performance of intel sgx. In *13th Web Information Systems and Applications Conference (WISA)*, pages 184–187. IEEE, 2016.

Variational Temporal IRT: Fast, Accurate, and Explainable Inference of Dynamic Learner Proficiency

Yunsung Kim
Stanford University
yunsung@stanford.edu

Sreechan Sankaranarayanan
Amazon.com LLC
sreeis@amazon.com

Chris Piech
Stanford University
piech@cs.stanford.edu

Candace Thille
Stanford University
cthille@stanford.edu

ABSTRACT

Dynamic Item Response Models extend the standard Item Response Theory (IRT) to capture temporal dynamics in learner ability. While these models have the potential to allow instructional systems to actively monitor the evolution of learner proficiency in real time, existing dynamic item response models rely on expensive inference algorithms that scale poorly to massive datasets. In this work, we propose Variational Temporal IRT (VTIRT) for fast and accurate inference of dynamic learner proficiency. VTIRT offers orders of magnitude speedup in inference runtime while still providing accurate inference. Moreover, the proposed algorithm is intrinsically interpretable by virtue of its modular design. When applied to 9 real student datasets, VTIRT consistently yields improvements in predicting future learner performance over other learner proficiency models.

Keywords

Item Response Theory, Dynamic IRT, Proficiency modeling, Variational Inference, Probabilistic Inference, Psychometric Models

1. INTRODUCTION

Evaluating the proficiency of a student is a fundamental task in education, and decades-long research in psychometrics have developed accurate probabilistic models to measure evidence of proficiency from student behaviors [17]. Item Response Theory (IRT) is the most well-known and widely applied probabilistic approach to proficiency modeling, which recognizes each response as a joint outcome of item features and student proficiency [19], and allows a single proficiency value per student to be estimated from responses to multiple assessment items.

However, in many routine aspects of educational practice, instructors and computer-based learning systems often use assessments more actively to assist learning rather than to evaluate learner proficiency post-hoc. Such assessments are referred to as *formative assessments* and are used not only to track student learning and make appropriate instructional interventions, but also to allow learners to practice their knowledge and skills, and make necessary self-corrections [17]. When learning occurs alongside assessment, learner proficiency is longitudinal rather than inert, and the assumption of static proficiency makes standard IRT less suitable as a model of proficiency measurement.

Dynamic Item Response models [14, 11] mitigate this issue by removing the assumptions of static ability and instead allowing it to stochastically change over time, but existing inference methods rely on expensive iterative algorithms with heavy runtime bottleneck. These methods scale poorly to massive datasets, which can be critical since in most use cases of dynamic proficiency modeling (e.g., learner proficiency monitoring), evaluation often needs to take place *real-time* to monitor the evolution of learner proficiency. This means that the expensive cost of inference must be incurred not just once, but multiple times over the course of a learner's learning experience.

In this paper, we develop Variational Temporal IRT (VTIRT), a fast and accurate framework for inferring dynamic learner proficiency over time. VTIRT is based on the idea of amortized variational inference [13], a fast approximate Bayesian inference framework for complex probabilistic models. The resulting algorithm infers the ability trajectory of a learner by first making *local* ability estimates in the form of a Gaussian distribution based on the item and response at each timestep (which we call the "*ability potentials*"), then aggregating these ability estimates across time in an intuitive fashion. In particular, our work delivers the following key innovations[1]:

- **Interpretable Inference for Dynamic IRT.** VTIRT allows the use of a structured probabilistic inference algorithm for sequence models through the notion of *ability potentials*, a form of conjugate potentials described in [12]. We concretely derive VTIRT in detail

Y. Kim, S. Sankaranarayanan, C. Piech, and C. Thille. Variational temporal irt: Fast, accurate, and explainable inference of dynamic learner proficiency. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 260–268, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115687

[1] Our public implementation of VTIRT based on PyTorch and Pyro [3] is available online in the following repository: `https://github.com/yunsungkim0908/vtirt`

and discuss the explainability of each of its components.

- **Fast and Accurate Inference.** Our proposed inference algorithm yields orders of magnitude speedup in inference runtime compared to existing inference algorithms while maintaining accurate inference.

- **Applications to Real World Datasets.** We apply our inference algorithm to 9 real student datasets. VTIRT consistently yields improvements in predicting future learner performance compared to other existing proficiency models.

2. RELATED WORKS

Many studies [20, 18, 7, 21, 22, 14] have investigated dynamic extensions of IRT that allow learner proficiency to vary over time. A common structure shared by these approaches is that student ability is assumed to follow a random walk:

$$\theta_{\ell,t} = \theta_{\ell,t-1} + \varepsilon_{\ell,t},$$

where $\varepsilon_{\ell,t}$ models a stochastic change in ability (often a zero-mean Gaussian). [7] finds a coarse approximation to the posterior distribution of per-time-step ability by ignoring the cross-temporal dependencies in the likelihood function while assuming knowledge of the item parameters. [14] and [21] use Markov Chain Monte Carlo (MCMC) methods [4] to estimate the unknown ability and item parameters. These methods draw samples asymptotically from the true posterior distribution conditioned on the observed responses, but the convergence of MCMC can be slow. On the other hand, [11] and [22] use Expectation-Minimization (EM) to iteratively estimate the dynamic item response parameters. In particular, [11] uses variational EM (VEM) to estimate the parameters of a distribution that closely approximates the true posterior distribution conditioned on the observed response. Although generally faster than MCMC-based methods, VEM methods still require costly iterative updates.

Closely related to the task of dynamic proficiency modeling is *knowledge tracing* [6, 16], which attempts to trace the knowledge of learners over time and accurately predict future performance. While Markov chain-based methods such as BKT [6] allow proficiency to be numerically measured through the estimated probability of being at a "proficient" state, the knowledge state representations of neural network-based knowledge tracing models [16] are not readily comparable or interpretable. Logistic regression knowledge tracing models offer simple and interpretable alternatives to neural network-based models. BestLR [9] and LKT [15] belong to this family of methods and use the number of correct and incorrect attempts as input features, while DAS3H [5] additionally embeds explicit representations of learning and forgetting over spans of time. VTIRT produces numerical representations of learner proficiency that are comparable by design across learners and across time, and its interpretable inference is also sensitive to the features of the attempted items.

Amortized variational inference has been used in [24] to develop VIBO for standard IRT. VIBO and its relationship to VTIRT are further discussed in Section 4.4.

3. VARIATIONAL INFERENCE REVIEW

Variational inference is a Bayesian framework for efficiently inferring unobserved variables in complex probabilistic models. In this setting, observations are modeled as samples from some underlying probability distribution (called the *generative model*) where some of the random variables (denoted r) are observed, and the remaining *latent* variables (denoted z) are unobserved. The goal of Bayesian inference then is to infer the latent random variables by finding the posterior distribution $p(z|r)$ given our knowledge of the likelihood distribution $p(r|z)$ and the prior distribution $p(z)$. This has the effect of "updating" the prior belief $p(z)$ with the observations to obtain the posterior belief $p(z|r)$.

For complex generative models, the posterior distribution $p(z|r)$ is often intractable to compute exactly. Variational inference is one way of doing *approximate* posterior inference that treats inference as an optimization problem, where we find the distribution $q(z)$ that is closest to the true posterior $p(z|r)$ from a more constrained (yet rich) family of distributions $\mathcal{Q}$ of our choice. This is achieved by maximizing an objective called "Evidence Lower BOund" (ELBO) for the observation r with respect to q

$$\mathcal{L}(q) \triangleq \mathbb{E}_{q(z)}\left[\frac{\log p(r|z)p(z)}{\log q(z)}\right], \tag{1}$$

which is equivalent to minimizing the Kullback-Leibler divergence between $q(z)$ and $p(z|r)$[2] due to the following equality:

$$\mathcal{L}(q) + KL\left(q(z)\|p(z|r)\right) = \log p(r) \equiv \text{Constant w.r.t } q.$$

Amortized Inference. What we just described is how VI works for a single observation. If we have a set of multiple i.i.d. observations sampled from the data-generating distribution $p_{\mathcal{D}}$ (which will be equal to the marginal distribution $p(r)$ if our generative model is correctly chosen), then finding the approximate posterior is equivalent to the following optimization problem

$$\arg\max_{q} \mathcal{L}(q) \triangleq \mathbb{E}_{p_{\mathcal{D}}(r)}\left[\mathbb{E}_{q_r(z)}\left[\frac{\log p(r,z)}{\log q_r(z)}\right]\right] \tag{2}$$

where we find one variational posterior factor q_r for each observation r. As the number of observations grows, however, finding q_r for each observation can quickly become highly inefficient. *Amortized Variational Inference* [8] tries to avoid this issue by *learning a mapping* $\phi(r)$ (also called the "recognition model") that maps observations to the parameters of the corresponding posterior distribution, rather than inferring each approximate posterior on the fly. By training a good recognition model ahead of time based on data and using it to retrieve the posterior distribution almost instantaneously at inference time, the cost of per-observation inference can be *amortized* [8]. Now we can choose the recognition model from a highly expressive family of functions (e.g., a neural network) and optimize the recognition model

[2] In fact, if $\mathcal{Q}$ includes the true posterior, then the q that achieves optimality will exactly be the the true posterior.

instead:

$$\arg\max_{\phi} \mathcal{L}(\phi) \triangleq \arg\max_{\phi} \mathbb{E}_{p_{\mathcal{D}}(r)} \left[\mathbb{E}_{q_{\phi(r)}(z)} \left[\frac{\log p(r,z)}{\log q_{\phi(r)}(z)} \right] \right]. \tag{3}$$

4. THE VTIRT FRAMEWORK

Based on the ideas of variational inference introduced earlier, we are now ready to describe the generative model and the inference algorithm that together comprise the VTIRT framework. The main intuition behind VTIRT's generative model is to incorporate temporality into IRT in a way similar to [7, 23]. Our framework, however, offers the additional flexibility to use *any* form of the item characteristic function - potentially with learnable parameters - whereas prior methods are constrained to a specific functional form.

4.1 The Temporal Ability Model

In our generative model (Figure 1a), we assume that the response $r_{\ell,t}$ of learner ℓ at timestep t is determined by 2-parameter IRT,

$$p\left(r_{\ell,t} | \theta, a, d\right) = f\left(a_{q_{\ell,t}}\left(\theta_{\ell,t} - d_{q_{\ell,t}}\right)\right), \tag{4}$$

where $q_{\ell,t}$ denotes the assessment item, $\theta_{\ell,t} \in [-\infty, \infty]$ denotes the ability of learner ℓ at timestep t, a_q and d_q each denote the discrimination and difficulty of assessment item q, and f denotes the linking function. To infuse temporality, we take an approach similar to [7, 23] and impose an additional assumption that a learner's ability is sampled from a random walk with Gaussian noise, also called a Wiener Process:

$$\theta_{\ell,t+1} | \theta_{\ell,t} \sim \mathcal{N}(\theta_{\ell,t}, \sigma_\theta^2), \quad \theta_{\ell,0} \sim \mathcal{N}(0, \sigma_\theta^2).$$

This is an instance of a more general Linear Gaussian model (LGM)

$$\theta_{\ell,t+1} | \theta_{\ell,t} \sim \mathcal{N}(\alpha_{\ell,t} \cdot \theta_{\ell,t} + \beta_{\ell,t}, s_{\ell,t}) \tag{5}$$

where the scale, bias, and standard deviation parameters are set to $(\alpha_{\ell,t}, \beta_{\ell,t}, s_{\ell,t}) = (1, 0, \sigma_\theta)$.[3]

The most popular choice for the linking function is the sigmoid function for 2 parameter logistic (2PL) IRT and Gaussian CDF for 2 parameter O-give (2PO) IRT. We will use 2PL as our modeling choice in our experiments considering its popularity [19]. It is important to note, however, that VTIRT makes *no assumption* about the linking function f as long as f is differentiable. Moreover, we can straightforwardly extend the model to admit a parameterized custom linking function f_ψ which we can learn from data. A similar approach in [24] has proven to yield better fit and higher predictive performance in the case of standard IRT, and we leave this extension to future research. This is in contrast to prior algorithms [7, 23] that become intractable for any linking functions other than a Gaussian CDF.

4.2 Choosing the Variational Family $\mathcal{Q}$

To do inference on our generative model, we first need to choose the variational family $\mathcal{Q}$. We will choose $\mathcal{Q}$ to be the family of distributions that factorize as follows:

$$q(\xi, \theta; r) = q(\xi) q(\theta | \xi, r) = \left(\prod_q q(\xi_q) \right) \left(\prod_\ell q(\theta_\ell | \xi, r) \right), \tag{6}$$

where we have used the shorthand notation $\xi_q = (a_q, d_q)$ to denote the features of the assessment item q. Since we are interested in inferring the temporal trajectory of abilities, we will choose $q(\theta|\xi, r)$ to be a Linear Gaussian Model just as its prior $p(\theta)$, and also choose $q(\xi)$ to be Gaussian. More precisely, we define $q(\theta|\xi, r)$ such that

$$\theta_{\ell,t+1} | \theta_{\ell,t}, \xi, r_\ell \sim \mathcal{N}\left(\alpha_{\ell,t} \cdot \theta_{\ell,t} + \beta_{\ell,t}, s_{\ell,t}\right) \tag{7}$$

whose scale $\alpha_{\ell,t}$, bias $\beta_{\ell,t}$, and standard deviation $s_{\ell,t}$ parameters are dependent on ξ and r_ℓ. Recalling the variational lower bound from Equation (1), our objective becomes

$$\mathcal{L}(q) = \mathbb{E}_{q(\xi)q(\theta|\xi,r)} \left[\frac{p(\xi)p(\theta)p(r|\xi,\theta)}{q(\xi)q(\theta|\xi,r)} \right]. \tag{8}$$

Since the parameters α_ℓ, β_ℓ and s_ℓ are dependent on the item parameters ξ and observed responses r_ℓ, it is tempting to apply the idea of amortized inference from Section 3 directly and model these parameters using learnable mappings. One such approach that we call **VTIRT_{dir-loc}** is to map the transition parameters at each timestep $1 \leq t \leq T$ based on the item parameters and responses from that timestep

$$\alpha_{\ell,t}, \beta_{\ell,t}, s_{\ell,t} = \phi\left(\xi_{q_{\ell,t}}, r_{\ell,t}\right). \tag{9}$$

While this approach is modular and its recognition model is low-dimensional and visualizable, its parameter estimates are not allowed to depend on responses *through time*, which may produce sub-optimal fit as we will later demonstrate through experiments. To allow dependence through time, we could instead choose to use a sequence-to-sequence recognition network (such as an LSTM network) to estimate the parameters for all time-steps at once using the entire sequence of responses:

$$\alpha_{\ell,1:T}, \beta_{\ell,1:T}, s_{\ell,1:T} = \phi\left(\xi_{q_{\ell,1:T}}, r_{\ell,1:T}\right). \tag{10}$$

We call this approach **VTIRT_{dir-s2s}**. While this uses a more expressive mapping, the increased complexity comes at the cost of interpretability and potentially a greater demand for more training data and long input sequences.

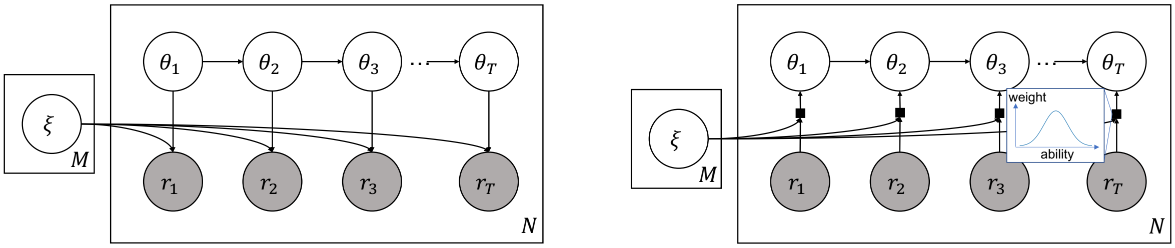
To mitigate this trade-off, we instead opt for an approach that is both modular enough to yield interpretability and yet also allows parameter estimates to depend on the responses through time.

4.3 VTIRT's Inference Algorithm

To describe our main inference method **VTIRT**, we first draw our attention to the following property about Linear Gaussian Models and Wiener processes, which will be foundational to our proposed method (See Appendix A for the proof):

Theorem 1. *Let $p(\theta_{1:T})$ be a Wiener process with standard deviation σ_θ and $q(\theta_{1:T})$ be a probability distribution*

[3] To allow for a fully Bayesian treatment, we also impose a Gaussian prior distribution on the item parameters: $a_q \sim \mathcal{N}(1, \sigma_a^2)$, and $d_q \sim \mathcal{N}(0, \sigma_d^2)$.



(a) VTIRT's Generative Model (b) VTIRT's Inference Model

Figure 1: Graphical model view of VTIRT's generative model and inference model. Shaded nodes indicate observed variables, and arrows denote the direction of dependence. Squares denote the ability potentials in the form of a Gaussian density.

defined as

$$q(\theta_{1:T}) \propto p(\theta_{1:T}) \prod_{t=1}^{T} \exp\left\{ \left(\frac{\theta_t - \mu_t}{\sigma_t} \right)^2 \right\}, \tag{11}$$

for real numbers $\mu_{1,\ldots,T}$ and $\sigma_{1,\ldots,T}$.

Then, $q(\theta_{1:T})$ is a Linear Gaussian Model[4]

$$\theta_t | \theta_{t-1} \sim \mathcal{N}(\widetilde{\mu}_t, \widetilde{\sigma}_t) \tag{12}$$

with

$$\widetilde{\mu}_t = \left(\frac{\lambda_\theta \theta_{t-1} + \lambda_t \mu_t + (\rho_{t+1}\lambda_\theta)\tau_{t+1}}{\lambda_\theta + \lambda_t + (\rho_{t+1}\lambda_\theta)} \right) \tag{13}$$

and

$$\widetilde{\sigma}_t = \sigma_\theta \sqrt{1 - \rho_{t+1}}, \tag{14}$$

where $\lambda_\theta = 1/\sigma_\theta^2$ and $\lambda_t = 1/\sigma_t^2$ denote precisions and parameters ρ_t and τ_t are defined recursively as

$$\rho_t = \left(\frac{\lambda_t + (\rho_{t+1}\lambda_\theta)}{\lambda_\theta + \lambda_t + (\rho_{t+1}\lambda_\theta)} \right), \rho_{T+1} = 0 \tag{15}$$

and

$$\tau_t = \left(\frac{\lambda_t \mu_t + (\rho_{t+1}\lambda_\theta)\tau_{t+1}}{\lambda_t + (\rho_{t+1}\lambda_\theta)} \right), \tau_{T+1} = 0. \tag{16}$$

In Equation (11), we are defining q by attaching *local "ability potentials"* to the prior distribution p, where each potential term is in the form of a Gaussian density with mean μ_t and variance σ_t^2. These potentials could be understood as local "beliefs" about the ability in the form of Gaussian distributions, judged solely based on the item features and learner response at the current timestep.

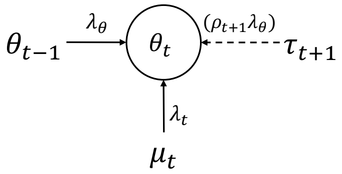


Figure 2: Schematic of VTIRT's inference at each timestep.

These potentials are combined across time with the prior distribution $p(\theta)$. The resulting θ_t follows a Gaussian distribution whose mean is a *weighted average* of the following 3 values that each represent information from different points in time (Figure 2): (1) θ_{t-1} of the previous timestep, (2) the local potential mean μ_t of the current timestep, and (3) the "future potential aggregate" τ_{t+1} that recursively aggregates potentials backwards from future timesteps via weighted averaging (Equation (16)). Each value is weighted proportionally to the *precision* (or "inverse uncertainty") associated with it[5], so the term with the lowest uncertainty contributes most to the resulting mean.

Therefore, Theorem 1 suggests a way to aggregate local ability estimates (Gaussian ability potentials) across timesteps using the global prior structure of the generative model. This motivates us to choose the following family of distributions for our variational factor $q(\theta)$ (Figure 1b):

$$q(\theta_\ell) \propto p(\theta_\ell) \prod_t \exp\left\{ \left(\frac{\theta_{\ell,t} - \mu(\xi_{q_{\ell,t}}, r_{\ell,t})}{\sigma(\xi_{q_{\ell,t}}, r_{\ell,t})} \right)^2 \right\}, \tag{17}$$

where $\mu(\cdot,\cdot)$ and $\sigma(\cdot,\cdot)$ are parameterized functions (e.g., feed-forward neural networks) that play the role of the recognition model. We refer to the resulting inference algorithm as **VTIRT**.

4.4 Conjugate Potentials and Variational IRT

VTIRT can be considered as a special case of using conjugate potential functions [12] to conduct approximate Bayesian inference, which allows intuitive and efficient inference algorithms designed for conditionally conjugate models to be used even when the model violates conjugacy. Specifically, the ability potentials in VTIRT enable efficient computation of variational posterior factors using a fast forward-backward inference algorithm for Linear Gaussian Models outlined in Theorem 1.

VIBO [24], an amortized variational inference algorithm for standard IRT, also belongs to this family of methods. In VIBO, the variational posterior distribution for ability is a Product-of-Experts where each "expert" component is a Gaussian distribution that depends locally on the response and item parameters from each timestep. These "experts" are also a form of conjugate potentials that allow variational posterior factors to be computed in closed-form.

[4] For notational convenience, we will use $\theta_0 = 0$

[5] $\rho_t\lambda_\theta$ can be viewed as the *effective precision* of the information coming from future timesteps.

Table 1: Statistics of the Workspace Learning Dataset

Course Name	Items	Learners	Interactions
Interviewing 1	89	79,808	5,458,576
Interviewing 2	12	10,536	120,388
Design Thinking	12	45,369	458,232
Software Development	8	10,277	80,137
Document Writing	13	20,043	233,175
Management A-1	28	10,154	247,674
Management A-2	16	14,673	198,720
Management B-1	14	21,293	281,844
Management B-2	14	15,254	206,108

This leads to several commonalities in both frameworks. Both use the same set of learnable parameters - the Gaussian posterior parameters (μ_{a_q}, μ_{d_q}, $\sigma^2_{a_q}$, $\sigma^2_{d_q}$) for each item q, and two recognition function components $\mu(\cdot,\cdot)$ and $\sigma(\cdot,\cdot)$ - and make inference by aggregating local ability potentials. While VIBO aggregates the conjugate potentials into a single univariate distribution over ability through a Product-of-Experts, VTIRT aggregates them into a Linear Gaussian Model based on Theorem 1. In Section 5, we will demonstrate through experiments that this difference in aggregation leads to VTIRT's performance improvement.

5. EVALUATION

We will now demonstrate that VTIRT achieves orders of magnitude faster inference than existing methods without compromising inference quality while also providing an interpretable structure. Experiments with real student data will also demonstrate that VTIRT yields a better fit to student behaviors than other learner proficiency models. We first describe the 2 datasets we used for our experiments.

5.1 Datasets

5.1.1 Synthetic Dataset

Using a simulated dataset enables us to test our algorithm under various hypothetical circumstances. We use VTIRT's generative model to simulate a set of learners responding to assessment items in an arbitrary order. For each learner, we first choose a random permutation of assessment items to simulate learners responding to assessment items in arbitrary order. Responses to these items are sampled based on the generative model defined in Section 4.1. This gives us access to the ground-truth item features and ability values that are otherwise unobtainable in real-world datasets. We set $\sigma_\theta = 0.25$ and $\sigma_a = \sigma_d = 1$ and vary the number of learners and the number of items.

5.1.2 Real Student Dataset: Workplace Learning

This dataset contains anonymized learner responses to a series of assessment questions in workplace learning courses taken by employees of a company. Each interaction record consists of (1) the ID of the assessment item (question), (2) ID of the learner, (3) correctness of the attempt, and (4) the knowledge components[6] with which each assessment item is associated (of which there could be multiple). Learners with fewer than 5 interactions throughout the course were omitted, and if there were multiple attempts to a question, only the first attempts were retained. A set of summary statistics for this dataset is presented in Table 1.

5.2 Fast and Accurate Inference

The most important quality of an inference algorithm is its capacity to promptly and reliably recover the unobserved variables based on past observations. The synthetic dataset allows us to measure this by comparing the computational runtime of a single instance of inference and computing the correlation of the inferred ability and item features against the known ground-truth values.

We implemented the 3 variants of VTIRT (VTIRT$_{\text{dir-loc}}$, VTIRT$_{\text{dir-s2s}}$, and VTIRT) along with 3 existing baseline inference methods - Variational EM (VEM), MCMC using Hamiltonian Monte Carlo[7] (HMC), and TSKIRT [7] - and used these algorithm to recover the latent ability values and item features for all learners and trials, based on the responses from all timesteps. (See Appendix B for more details about the methods and the experiment.) We varied the number of items from 32 to 500 while fixing the number of learners to 5000, then varied the number of learners from 2,500 to 20,000 while fixing the number of items to 250.

Figure 3 plots the inference time and Pearson correlations of the model estimates with the ground-truth values. Most notably, all 3 variants of VTIRT are *orders of magnitude* faster than other inference methods. Moreover, VTIRT consistently yields the best discrimination estimates. Except when there are few items, the difference in the quality of ability and difficulty estimates are also minor compared to VEM (up to 0.07 difference in ability correlation and 0.03 difference in difficulty correlation).

Among all variants of VTIRT, VTIRT using ability potentials consistently outperforms direct amortization. As noted earlier, VTIRT$_{\text{dir-loc}}$ ignores temporal dependency in estimating the transition dynamics, while the complexity of VTIRT$_{\text{dir-s2s}}$ could come at the cost of the need for more training data and long input sequences.

5.3 Application to Real Student Data

We now compare VTIRT with other proficiency models in modeling real student data. Since we do not have access to the ground-truth learner ability in reality, our evaluation on real student data must be based on a related proxy metric. As a proxy, we will focus on the task of predicting the *next step* response correctness of learners based on the model's *current* ability estimates and item features.[8]

We compared the predictive performance of VTIRT against the following baseline: IRT, BKT, VIBO[24][9], VTIRT$_{\text{dir-loc}}$,

[6] Most courses had 2-4 knowledge components.

[7] Hamiltonian Monte Carlo [2, 10] is an efficient MCMC algorithm for continuous state spaces.

[8] Since the items in each course were associated with different knowledge components, we estimated learner ability for each knowledge component separately. Prediction on each item was made based on the ability averaged across the knowledge components associated with that item.

[9] To adopt VIBO to a sequential estimation setting, we computed the ability estimates at each timestep separately using the responses prior to that timestep.

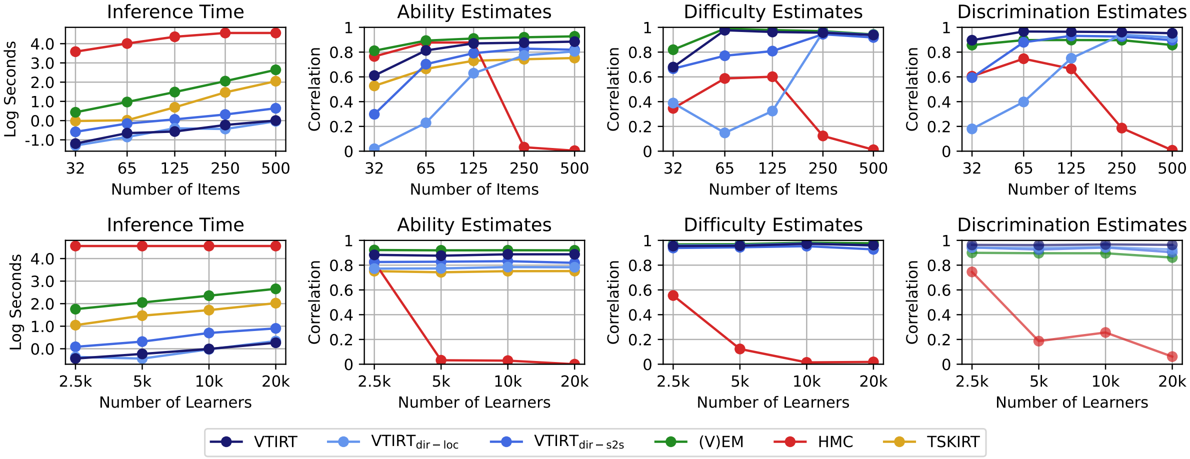


Figure 3: Performance on the synthetic dataset. Inference time was capped at 10 hours.

Table 2: Next-Step Performance Prediction ROC.

	IRT	BKT	VIBO	VTIRT	VTIRT$_{\text{dir-loc}}$	VTIRT$_{\text{dir-s2s}}$	VTIRT$_{\text{transfer}}$
Interviewing 1	0.702	0.622	0.752	**0.762**	0.758	0.749	0.756
Interviewing 2	0.586	0.632	0.765	**0.779**	0.774	0.772	0.760
Software Development	0.565	0.648	0.701	**0.711**	0.695	0.667	0.702
Design Thinking	0.602	0.605	0.674	**0.681**	0.677	0.646	0.633
Document Writing	0.503	0.683	0.754	**0.770**	0.766	0.750	0.746
Management A-1	0.518	0.639	0.717	**0.738**	0.734	0.729	0.723
Management A-2	0.705	0.682	0.771	**0.774**	0.770	0.766	0.770
Management B-1	0.570	0.582	0.734	**0.741**	0.739	0.730	0.735
Management B-2	0.733	0.602	0.766	**0.770**	0.766	0.765	0.766

and VTIRT$_{\text{dir-s2s}}$.[10] To study the effect of VTIRT's forward-backward inference algorithm, we also analyzed the performance of a variant of VTIRT we call **VTIRT$_{\text{transfer}}$** in which we train the recognition networks using VIBO and perform inference using VTIRT's inference algorithm.

Table 2 reports the average AUROC on this prediction task over a 5-fold cross-validation, where the learners were split into 5 equally-sized splits. These results suggest the following observations:

VTIRT consistently outperforms other proficiency models. VTIRT achieves up to 2.1 AUROC point advantage in comparison to the best performing baseline, VIBO. As VIBO and VTIRT share the same parameterization scheme, the increased performance is attributable to the VTIRT framework.

Ability potentials are more effective than direct amortization. VTIRT using ability potentials outperforms both the local and sequence-to-sequence direct amortization variants. It is interesting to note that local direct amortization also outperformed LSTM-based sequence-to-sequence direct amortization in all courses, which may be due to relatively short sequence length per knowledge component.

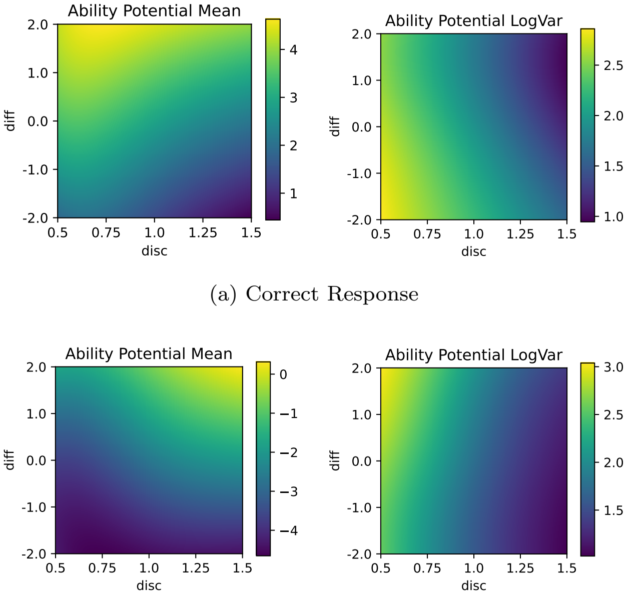
VTIRT's training mechanism is critical to its performance. Since VTIRT and VIBO have the same parameterization schemes, it is natural to ask whether VTIRT's sequential training could be replaced with VIBO's parallelizable training without much loss in performance. Comparing the performance of VTIRT$_{\text{transfer}}$ with VTIRT, we see that VTIRT's training mechanism is crucial to the enhanced performance, and VTIRT$_{\text{transfer}}$ often performs far worse than VIBO itself.

5.4 Interpretability of VTIRT

VTIRT is a modular algorithm, and by virtue of its structure, all parts of its operations are intrinsically interpretable. The ability estimates are computed from the local ability potentials, following the logic outlined in Section 4.3. These ability potentials provide "local beliefs" of the learner's ability at each timestep in the form of a Gaussian distribution and are aggregated through the forward-backward inference algorithm based on Theorem 1.

One of the merits of this potential function is that its dimensions are low enough to be visually analyzed. Figure 4 is a

[10] We used the popular MIRT package in R for the IRT baseline, and the implementation from the pyBKT package [1] for the BKT baseline. Since VTIRT and VIBO's estimates take the form of a probability distribution, we used the mean of the distribution as the model's point-estimate and fed it as input to the 2PL IRT likelihood function in Equation (4) to compute the predicted probability of correctness.



(a) Correct Response

(b) Incorrect Response

Figure 4: Mean and log variance of the ability potentials as a function of the item correctness and item features.

plot of the mean and log variance[11] of the potential function for the "Interviewing 2" course for typical parameter ranges, and its shape aligns with our intuitive expectations of how a learner's response would affect our belief of its ability depending on the item features. In particular,

- For assessment items of any difficulty and discrimination, a correct response always yields higher ability estimate than an incorrect response (which can be seen from the range of the color bar).
- The uncertainty of the ability estimates are generally lower (so the model is more certain about its estimates) for items with higher discrimination. This aligns with the expectation that high discrimination items are useful for distinguishing learners with different abilities.
- Correct responses to high-difficulty items yield potentials with greater mean and lower uncertainty than correct responses to low-difficulty questions (and the opposite for incorrect responses).[12]

6. LIMITATIONS AND FUTURE WORK

Adaptive and Self-Directed Learning. The key characteristic of VTIRT is its ability to make sequential ability estimates from responses to a set of heterogeneous assessment items. For this reason, we hypothesize that the ideal environment for VTIRT in comparison to other proficiency models is one where learners possess great agency in choosing their learning trajectories, or where the learning trajectories are adjusted adaptively to the performance of the learner. However, most learners in our real student dataset followed similar learning trajectories with little variability, and this hypothesis remains untested. An important direction for future work would be to test our framework in an adaptive or self-directed learning environment.

Modeling Assumptions of VTIRT. One interesting topic for future research is the modeling assumption made by VTIRT. VTIRT's generative model builds on a simple assumption that learner ability starts close to 0 and that the changes in ability are Gaussian with mean 0. Under this generative model, the temporal changes in ability may take on both positive and negative values. While we have shown using real student data that the resulting inference algorithm yields a more accurate fit, research remains to be done to examine how the modeling assumptions could be further improved.

Ability Potential for Atypical Item Parameter Values. In Section 5.4, we visualized in Figure 4 the trained ability potential function for one of the datasets for typical ranges of the item parameter values. Yet, the input to the potential function can be any tuple $\xi = (a, d)$ of unbounded real numbers, and the typical range of input observed during training comprise only a very small subset of this domain. For values of the item parameters outside this typical range, the trained potential function may fail to generalize as a result of sparse training signal and exhibit arbitrary behaviors. Enhancing the generalizability of the potential function and its robustness to extreme values of the item parameters is an exciting direction for future research.

Logistic Regression Knowledge Tracing Models. Logistic regression models of knowledge tracing such as BestLR [9] or LKT [15] share several similarities with VTIRT. As noted earlier, these models use the number of correct and incorrect past attempts in a learning trajectory to predict future performance, and VTIRT makes inference on ability based on both the historical performance of the learner and the features of the attempted items. While the focus of this study was to develop scalable inference for dynamic IRT models and compare the model fit against other proficiency models, it remains an interesting future research to compare VTIRT against logistic regression knowledge tracing models under both adaptive and non-adaptive learning environments.

7. CONCLUSION

We presented VTIRT, a fast and accurate inference framework for dynamic item response models. VTIRT offers orders of magnitude speedup in the inference runtime while maintaining a highly accurate inference of learner and item parameters. Moreover, every component of our inference algorithm is interpretable by virtue of its modular design. Experiments on real student data demonstrates that VTIRT achieves improvements in inferring future learner performance compared to other proficiency models.

[11] High variance indicates large uncertainty.

[12] Although it may seem as if correct responses to low-discrimination items yield higher ability estimates because the mean parameter is greater, the overall distribution is in fact flatter and more spread out in general due to higher variance.

8. REFERENCES

[1] A. Badrinath, F. Wang, and Z. Pardos. pybkt: An accessible python library of bayesian knowledge tracing models. *International Educational Data Mining Society*, 2021.

[2] M. Betancourt. A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*, 2017.

[3] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. D. Goodman. Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research*, 20(1):973–978, 2019.

[4] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. *Handbook of markov chain monte carlo.* CRC press, 2011.

[5] B. Choffin, F. Popineau, Y. Bourda, and J.-J. Vie. Das3h: Modeling student learning and forgetting for optimally scheduling distributed practice of skills. In *International Conference on Educational Data Mining (EDM 2019)*, 2019.

[6] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[7] C. Ekanadham and Y. Karklin. T-skirt: Online estimation of student proficiency in an adaptive learning system. *arXiv preprint arXiv:1702.04282*, 2017.

[8] S. Gershman and N. Goodman. Amortized inference in probabilistic reasoning. In *Proceedings of the annual meeting of the cognitive science society*, volume 36, 2014.

[9] T. Gervet, K. Koedinger, J. Schneider, T. Mitchell, et al. When is deep learning the best approach to knowledge tracing? *Journal of Educational Data Mining*, 12(3):31–54, 2020.

[10] M. D. Hoffman, A. Gelman, et al. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

[11] K. Imai, J. Lo, and J. Olmsted. Fast estimation of ideal points with massive data. *American Political Science Review*, 110(4):631–656, 2016.

[12] M. J. Johnson, D. K. Duvenaud, A. Wiltschko, R. P. Adams, and S. R. Datta. Composing graphical models with neural networks for structured representations and fast inference. *Advances in neural information processing systems*, 29, 2016.

[13] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[14] A. D. Martin and K. M. Quinn. Dynamic ideal point estimation via markov chain monte carlo for the us supreme court, 1953–1999. *Political analysis*, 10(2):134–153, 2002.

[15] P. I. Pavlik, L. G. Eglington, and L. M. Harrell-Williams. Logistic knowledge tracing: A constrained framework for learner modeling. *IEEE Transactions on Learning Technologies*, 14(5):624–639, 2021.

[16] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. *Advances in neural information processing systems*, 28, 2015.

[17] R. K. Sawyer. *The Cambridge handbook of the learning sciences.* Cambridge University Press, 2005.

[18] C. Studer. Incorporating learning over time into the cognitive assessment framework. *Unpublished PhD, Carnegie Mellon University, Pittsburgh, PA*, 2012.

[19] W. J. Van der Linden and R. Hambleton. Handbook of item response theory. *Taylor & Francis Group. Citado na pág*, 1(7):8, 1997.

[20] P. Van Rijn et al. Categorical time series in psychological measurement. *Psychometrika*, 62:215–236, 2008.

[21] X. Wang, J. O. Berger, and D. S. Burdick. Bayesian analysis of dynamic item response models in educational testing. *The Annals of Applied Statistics*, 7(1):126–153, 2013.

[22] R. C.-H. Weng and D. S. Coad. Real-time bayesian parameter estimation for item response models. *Bayesian Analysis*, 13(1):115–137, 2018.

[23] K. H. Wilson, Y. Karklin, B. Han, and C. Ekanadham. Back to the basics: Bayesian extensions of irt outperform neural networks for proficiency estimation. *arXiv preprint arXiv:1604.02336*, 2016.

[24] M. Wu, R. L. Davis, B. W. Domingue, C. Piech, and N. Goodman. Variational item response theory: Fast, accurate, and expressive. *arXiv preprint arXiv:2002.00276*, 2020.

APPENDIX

A. PROOF OF THEOREM 1

We will first find the parameters α_t, β_t, s_t of the resulting Linear Gaussian Model (Equation (5)) by solving for the following equation:

$$\begin{aligned}
&\log q(\theta_{1:T}) \\
&= \left(\frac{\theta_1}{\sigma_\theta}\right)^2 + \sum_{t=1}^{T} \left\{ \left(\frac{\theta_t - \theta_{t-1}}{\sigma_\theta}\right)^2 + \left(\frac{\theta_t - \mu_t}{\sigma_t}\right)^2 \right\} + C \\
&= \left(\frac{\theta_1 - \beta_1}{s_1}\right)^2 + \sum_{t=2}^{T} \left(\frac{\theta_t - \alpha_t\theta_{t-1} - \beta_t}{s_t}\right)^2 + C', \qquad (18)
\end{aligned}$$

where C and C' are constants with respect to $\theta_{1:T}$. Rearranging terms and comparing the coefficints of the terms involving $\theta_t\theta_{t-1}$, we obtain

$$s_t = \sigma_\theta\sqrt{\alpha_t}.$$

Substituting this into Equation (18) and comparing the terms involving θ_t and θ_t^2, we obtain the following recursive system of equations:

$$\alpha_t = \frac{\lambda_\theta}{\lambda_\theta + \lambda_t + (1-\alpha_{t+1})\lambda_\theta},$$
$$\beta_t = \frac{\mu_t\lambda_t + \beta_{t+1}\lambda_\theta}{\lambda_\theta + \lambda_t + (1-\alpha_{t+1})\lambda_\theta},$$

where $\alpha_{T+1} = 1$ and $\beta_{T+1} = 0$ are defined for notational simplicity. Note from the above equation that

$$\frac{b_t}{1-\alpha_t} = \frac{\lambda_t + (1-\alpha_{t+1})\lambda_\theta}{\mu_t\lambda_t + (1-\alpha_{t+1})\lambda_\theta\left(\frac{\beta_{t+1}}{1-\alpha_{t+1}}\right)}.$$

This motivates us to define $\rho_t = 1-\alpha_t$ and $\tau_t = \frac{\beta_t}{1-\alpha_t}$, which yields the formula in Equations (15) and (16):

$$\rho_t = \left(\frac{\lambda_t + (\rho_{t+1}\lambda_\theta)}{\lambda_\theta + \lambda_t + (\rho_{t+1}\lambda_\theta)}\right), \; \tau_t = \left(\frac{\lambda_t\mu_t + (\rho_{t+1}\lambda_\theta)\tau_{t+1}}{\lambda_t + (\rho_{t+1}\lambda_\theta)}\right).$$

$\widetilde{\mu}_t$ in Equation (12) then satisfies

$$\begin{aligned}\widetilde{\mu}_t &= \alpha_t\theta_{t-1} + \beta_t = (1-\rho_t)\theta_{t-1} + \rho_t\tau_t \\ &= \left(\frac{\lambda_\theta\theta_{t-1}}{\lambda_\theta + \lambda_t + (\rho_{t+1}\lambda_\theta)}\right) + \left(\frac{\lambda_t\mu_t + (\rho_{t+1}\lambda_\theta)\tau_{t+1}}{\lambda_\theta + \lambda_t + (\rho_{t+1}\lambda_\theta)}\right) \\ &= \left(\frac{\lambda_\theta\theta_{t-1} + \lambda_t\mu_t + (\rho_{t+1}\lambda_\theta)\tau_{t+1}}{\lambda_\theta + \lambda_t + (\rho_{t+1}\lambda_\theta)}\right),\end{aligned}$$

and $\widetilde{\sigma}_t = s_t = \sigma_\theta\sqrt{a_t} = \sigma_\theta\sqrt{1-\rho_t}$.

B. EXPERIMENT DETAILS

For all implementation of the VTIRT variants, we used a 2-layer feedforward neural network with 16 dimensional hidden layers with GELU activation for the potential function.

While TSKIRT requires the item parameters to be learned in advance using standard IRT, we used the ground-truth item parameters instead of training the item parameters with a different model - all other algorithms had to infer the item parameters from scratch.

All experiments were run on identically configured CPU machines (2 AMD EPYC 7502 32-Core Processors and 10 gigabytes of memory) until convergence for a maximum of 10 hours, *with the exception of VEM*. VEM makes batch updates to the latent posterior estimates, and its item parameter updates can be significantly sped up through vectorized indexing. This speedup, however, incurs a large memory overhead. To make a conservative comparison of VTIRT's run time performance against the ideal setup for VEM, we applied this vectorization to VEM, but had to allow it to use *4 times* the memory allocated to other methods, especially for the larger datasets.

Session-based Course Recommendation Frameworks using Deep Learning

Md Akib Zabed Khan
Florida International University
mkhan149@fiu.edu

Agoritsa Polyzou
Florida International University
apolyzou@fiu.edu

ABSTRACT

Academic advising plays an important role in students' decision-making in higher education. Data-driven methods provide useful recommendations to students to help them with degree completion. Several course recommendation models have been proposed in the literature to recommend courses for the next semester. One aspect of the data that has yet to be explored is the suitability of the recommended courses taken together in a semester. Students may face more difficulty coping with the workload of courses if there is no relationship among courses taken within a semester. To address this problem, we propose to employ session-based approaches to recommend a set of courses for the next semester. In particular, we test two session-based recommendation models, CourseBEACON and CourseDREAM. Our experimental evaluation shows that session-based methods outperform existing popularity-based, sequential, and non-sequential recommendation approaches. Accurate course recommendation can lead to better student advising, which, in turn, can lead to better student performance, lower dropout rates, and better overall student experience and satisfaction.

Keywords

session-based recommendation, course recommendation, deep learning

1. INTRODUCTION

In higher education, one of the challenges that students face is identifying the proper set of courses to register for every semester so that they will successfully progress with their degree. While selecting courses, students consider different factors, like a balance between their personal preferences (interests, goals, and career aspirations) and the requirements of their degree programs. Students usually need to register for some elective courses. Some courses have prerequisites. These decisions are hard to make, and consequently, appropriate course selection is a non-trivial task for the students. Courses can be selected based on example guides provided by each department, but these are not adapted to individual cases [6]. Students may get personalized assistance from academic advisors. However, the ratio of students to advisors is very high, which limits discussion time between an advisor and an advisee [11]. Lack of communication may result in unfavorable situations where students fail to cope with the course workload and become frustrated. As a matter of fact, the dropout rates at the undergraduate level of US colleges are alarming [10].

Data mining techniques and machine learning models can draw insights from historical data records and generate course recommendations to assist with student advising. Collaborative filtering algorithms and content filtering methods are the most common approaches in this field of research. Existing work has explored the sequential nature of course enrollment data, the words associated with course description data, and non-sequential approaches to prioritize students' preferences and analyze their characteristics and skills to recommend courses for a semester or even multiple consecutive semesters. However, no prior study in the literature considers how well the courses would be suited to be taken together within a semester. Some courses are usually taken together if they cover complementary concepts. Also, it is not a good idea for a student to take some very heavy courses in the same semester. For example, if a student registers for three or four difficult courses in the same semester, that could lead to poor performance in some or all of them, as the student will not have enough time to study for all the courses. In the end, their semester grade point average (GPA) might be low compared to their efforts.

Students might be more likely to perform well if their courses are related and not so difficult to study altogether within a semester. The set of courses taken in a semester by past students can provide impactful insights to measure the correlation, relationship, and compatibility of a pair of courses. These notions can be captured by session-based recommendation approaches. We propose to adapt such approaches to rank courses not only based on their suitability but based on their synergy as well. There are long short-term dependencies in the sequence of courses taken semester-by-semester, and we can capture them by using deep learning models.

We explore two different models that capture these dependencies. First, we propose CourseBEACON, where we calculate the co-occurrence rate between a pair of courses to

M. A. Z. Khan and A. Polyzou. Session-based course recommendation frameworks using deep learning. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 269–277, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115699

capture and estimate their relationship. Then, we incorporate this notion of course compatibility into sequential deep learning models (recurrent neural networks) to perform the recommendation task. Second, in CourseDREAM, we create latent vector representation for each basket of courses taken in a semester which is helpful to capture the courses that are historically considered to be suitable to take within a semester. Then, we use recurrent neural networks (RNNs) to capture the sequential transition of courses over the sequence. Using real historical data, we evaluate these proposed approaches. They outperform other baselines or existing state-of-the-art approaches we examined. Our course recommendation model will be impactful in academic advising to help students with decision-making and decrease the risk of dropout cases. The paper is organized as follows. Sect. 2 introduces the background, notation, and related work, while Sect. 3 delves deeper into the two proposed approaches. Sect. 4 presents our experimental setup in detail. Sect. 5 shows our results and Sect. 6 concludes the paper.

2. BACKGROUND

2.1 Problem

Some courses are needed to fulfill degree requirements, others are elective. Usually, it is up to the students to decide which courses to take within a semester and in which semester they will take any required courses. While selecting courses, students must remember degree requirements and several factors such as personal preferences, course prerequisites, career goals, and which courses are needed to build knowledge for future courses. Universities naturally collect information about the course registration history of students. Insightful patterns can be extracted by analyzing the historical information of past students to recommend courses to future students. *Course recommendation* is a systematic way to evaluate which courses are appropriate for a student with the goal of making advising easier. By inspecting the student-course interactions, and sequential transitions of courses over the semesters of past students, we recommend courses for other students by implementing various data-driven techniques.

2.2 Definition of Terms and Notations

Considering the terminology used in general recommendation literature, we can consider each student as a user, and each course as an item.

A **session** is a finite amount of time for a user to complete a set of activities together. In this paper, we consider a student's semester to be a session. A user can buy a set of items together in a session. A student can take a set of courses together within a semester. A basket is a similar notion to a session. In **session-based recommendation** models, we learn users' preferences from the sessions generated during the consumption process and pay increasing attention to recent sessions as users' preferences change and evolve dynamically. A session-based recommendation system recommends a set of items for the next session, for which we may or may not have some partial information (i.e., any items already present in the session). A **next-basket recommendation** is a special case when we generate a complete set of items for the next session (i.e., without any partial information). In this paper, we will recommend a complete set of courses for the next semester.

Table 1: Notations

$\mathcal{C}$, $\mathcal{S}$	set of courses and students, respectively
m, n	cardinality of $\mathcal{C}$, $\mathcal{S}$, $\lvert\mathcal{C}\rvert = m$ and $\lvert\mathcal{S}\rvert = n$
p, q	courses, $p, q \in \mathcal{C}$
u	student, $u \in \mathcal{S}$
i, j	index for student and course, respectively
t	index for semester
$\mathcal{B}_{i,t}$	set of courses i-th student took in semester t
H_i	course registration history of i-th student
t_i	total number of semesters for i-th student, $t_i = \lvert H_i\rvert$
k	total number of courses to recommend
$\mathbf{R}$	tensor $\mathbf{R} \in \mathbb{R}^{d \times d \times n}$ with d latent dimensions
$\mathbf{F}_{i,p,q}$	number of (i, p, q) triples for i-th student

We adopt the following notation. We will use calligraphic letters for sets, capital bold letters for matrices or tensors, and lowercase bold letters for vectors. $\mathcal{C}$ indicates the set of all courses ($|\mathcal{C}| = m$) and $\mathcal{S}$ denotes the set of all students ($|\mathcal{S}| = n$). $\mathcal{B}_{i,t}$ represents a set of courses that i-th student has taken in a semester t. H_i is the course registration history of the i-th student, $H_i = [\mathcal{B}_{i,1}, \mathcal{B}_{i,2}, ..., \mathcal{B}_{i,t_i}]$, where $t_i = |H_i|$ is the total number of semesters that the i-th student took courses. Additional notation is presented in Table 1. We will refer to the student and the semester we are trying to generate a recommendation for as the target student and semester, respectively.

2.3 Related Work

Researchers have analyzed different types of data to build course recommendation models using different techniques. In terms of *types of data*, many researchers used real-life course enrollment and course description datasets collected from universities' data warehouses [1, 8, 22, 23, 27, 32], and others used online datasets collected from different online course platforms [5, 19, 34]. Moreover, some of the researchers collected data by taking feedback from students conducting surveys [8, 22]. Some researchers considered the grades of students in each course as a useful feature to recommend courses that students were expected to perform well [8, 15, 17, 19, 32]. Others did not consider grades as an important feature [1, 21, 22, 23, 34]. Very few researchers considered the interests and skills of students to choose a course [14, 28] by collecting students' survey responses.

Different *course recommendation systems* have been proposed in the literature. Parameswaran et al. introduced the first course recommendation system based on constraint satisfaction [20]. Al-Badarenah et al. propose a collaborative filtering-based course selection system using a k-means clustering algorithm and an association rule mining method [1]. The Apriori algorithm (an association rule mining technique) has been used for a collaborative recommendation system for online courses [19]. There are some content-based filtering methods available in literature where researchers analyze course descriptions and course content to recommend courses [17, 18, 21, 22]. Pardos et al. propose a course2vec model (like word2vec model) for course recommendation using both course enrollment and course description data [22]. Students' preferences and student-course interactions are neglected in these methods. To prioritize students' preferences, a matrix factorization model has been proposed for course recommendation [29]. Pardos et al. propose a combination of long short-term memory (LSTM) networks and skip-gram models to recommend courses balancing explicit and implicit preferences of students [21]. RNN models have also been used to recommend courses that are expected to improve students' GPAs [16]. Shao et al. introduce a PLAN-BERT model to recommend multiple consecutive semesters toward degree completion [27]. Polyzou et al. present a random-walk-based approach, Scholars Walk, to capture sequential transitions of different courses semester-by-semester assuming that the past sequence of courses matters [23]. No prior study captures the relationship among courses taken in a semester considering each semester as a session to recommend correlated courses.

In commercial recommender systems, there are different types of *session-based recommendation systems* to recommend the next clicked item (next interaction), the next partial session (subsequent part) in the current session, and the next basket or complete session with respect to the previous sessions for a user [31]. For our problem setting, the last one is more appropriate. Next-basket recommendation approaches can be divided into two types: sequential and non-sequential approaches. Generally, sequential approaches capture the user-item interactions and sequential relationships of items, by constructing a transition matrix observing item transitions over sessions for a user. Rendle et al. introduced the first next-basket recommendation system by presenting a factorized personalized Markov chain (FPMC) model [26]. The FPMC model can capture the first-order dependency of items. Long short-term dependency of items over the sequence of baskets can be captured by recurrent neural networks. Yu et al. propose a dynamic recurrent basket model (DREAM) using LSTM networks [33]. Le et al. propose a correlation-sensitive next basket recommendation model named Beacon to recommend correlated items using transaction data of online grocery shops [13].

Non-sequential approaches prioritize users' preferences of items over the sequential transition of items. Matrix factorization and tensor decomposition techniques have been used to recommend the next item capturing users' preferences of choosing one item over another item [7, 25]. Wan et al. propose a representation learning model triple2vec to recommend complementary and compatible items in the next basket [30]. A tensor decomposition technique has been proposed to recommend complementary items in the next basket by using RESCAL decomposition [7].

In this paper, we explore both sequential and non-sequential approaches for session-based recommendation to recommend a set of synergistic courses for the next semester. Moreover, our course enrollment data is different from transaction data of items, i.e., one item can appear multiple times in different sessions in a sequence of baskets for a user, but one course is most likely to appear one time in a sequence of semesters for a student.

3. SESSION-BASED COURSE RECOMMENDATION

We propose to use a session-based, sequential course recommendation system to capture the synergy between courses taken in the same semester. Even though some courses might be worth equal credit hours, the required working time load varies based on the difficulty of subjects [4]. Students' course load can impact their performance [2]. A good combination of courses can balance the workload of courses. The courses well-suited to be taken together could cover similar topics, be correlated, or just not be overwhelming for the students. The influence of co-taken courses has been considered important for other educational tasks, i.e., grade prediction and designing an early warning system [3, 9, 24].

We analyze the relationship and correlation of courses by incorporating the concept of session-based recommendation (SBR) for the first time in course recommendation. We consider a set of courses taken in a semester as a session and inspect the session to understand the relationship among the courses. Let $\mathcal{B}_{i,t}$ be a set of courses of the i-th student at semester t. Given the courses for the first $t_i - 1$ semesters for the i-th student as input, our target is to recommend a set of correlated courses, $c_1, c_2 ..., c_k$ for the target semester t_i where k is the number of courses to be recommended. We extend two popular session-based models, the Beacon model [13] to CourseBEACON, and DREAM model [33] to CourseDREAM, for the purpose of course recommendation.

3.1 Assumptions

We make the following assumptions in the context of course recommendations in higher education.

1. Time is discrete and moves from one semester to the next.
2. The courses are ordered over the sequence of semesters, but there is no order in the courses within a semester.
3. Learning is sequential; each course taken in a semester provides some skills and knowledge that will help to learn future courses in the following semesters. So, the sequence of courses matters in course selection.
4. Course registration history of students might provide beneficial insight into the curriculum and degree requirements when sufficient domain experts are not available.
5. We know the number of courses the student will take in the target semester.

3.2 CourseBEACON

We adopt the Beacon model to CourseBEACON to generate correlation-sensitive course recommendations for next semester. The framework has three components: correlation-sensitive basket encoder, course basket sequence encoder, and correlation-sensitive score predictor. The basket encoder receives as input the sequence of courses taken in the previous semesters $[1, ..., t_i - 1]$ and the global correlation matrix, $\mathbf{M} \in \mathbb{R}^{m \times m}$, which captures the relationships among courses within a semester (basket). The encoder produces a sequence of basket representations as output for each prior basket of a student. We feed these representations into the course basket sequence encoder where LSTM networks capture the sequential association of courses over the sequence of semesters. Finally, we feed the output from the sequence encoder and the correlation matrix as inputs to the correlation-sensitive score predictor to produce the final scores for the candidate courses. We recommend the courses with the highest score for each student.

Building the Course Correlation Matrix We create the correlation matrix using all semesters available in training data. Let $\mathbf{F} \in \mathbb{R}^{m \times m}$ be the frequency matrix. For courses $p, q \in \mathcal{C}$, $\forall p \neq q$, we count $\mathbf{F}_{p,q}$, which is the number of times p, q co-occur together in a basket (i.e., taken in the same semester). We normalize $\mathbf{F}$ to generate the final correlation matrix $\mathbf{M}$ based on the Laplacian matrix $\mathbf{M} = \mathbf{D}^{-1/2}\mathbf{F}\mathbf{D}^{-1/2}$, where $\mathbf{D}$ denotes the degree matrix, $D_{p,p} = \sum_{q \in \mathcal{C}} \mathbf{F}_{p,q}$ [12]. $\mathbf{F}$ and $\mathbf{M}$ are symmetric by definition.

Correlation-Sensitive Basket Encoder For each semester, we create a binary indicator vector for the set of courses that were taken by a student. We convert a basket of courses $\mathcal{B}_{i,t}$ to binary vector $\mathbf{b}_{i,t}$ of length m for the i-th student, where the j-th index is set to 1 if $c_j \in \mathcal{B}_{i,t}$; otherwise it is 0. We get an intermediate basket representation $\mathbf{z}_{i,t}$ for a basket $\mathcal{B}_{i,t}$ as follows:

$$\mathbf{z}_{i,t} = \mathbf{b}_{i,t} \odot \omega + \mathbf{b}_{i,t} * \mathbf{M}, \tag{1}$$

where ω is a learnable parameter that indicates the importance of the course basket representation, the circle-dot indicates element-wise product, and the asterisk indicates matrix multiplication. We feed $\mathbf{z}_{i,t}$ into a fully-connected layer and we get a latent basket representation $\mathbf{r}_{i,t}$ by applying the ReLU function in an element-wise manner:

$$\mathbf{r}_{i,t} = ReLU(\mathbf{z}_{i,t}\Phi + \phi), \tag{2}$$

where Φ, ϕ are the weight and bias parameters, respectively.

Course Basket Sequence Encoder We use the sequence of latent basket representations $\mathbf{r}_{i,t}, \forall t \in [1, \ldots, t_i - 1]$ for the i-th student as input in the sequence encoder. Each $\mathbf{r}_{i,t}$ is fed into an LSTM layer, along with the hidden output from the previous layer. We compute the hidden output $\mathbf{h}_{i,t}$ as:

$$\mathbf{h}_{i,t} = tanh(\mathbf{r}_{i,t}\Psi + \mathbf{h}_{\mathbf{i},(\mathbf{t-1})}\mathbf{\Psi'} + \psi), \tag{3}$$

where Ψ, Ψ' and ψ are learnable weight and bias parameters.

Correlation-Sensitive Score Predictor We use the correlation matrix and the last hidden output as the input in the correlation-sensitive score predictor to derive a score for each candidate course. Let $\mathbf{h}_{i,t_i-1}$ be the last hidden output generated from the sequence encoder. First, we get a sequential signal $\mathbf{s}_i$ from the given sequence of baskets:

$$\mathbf{s}_i = \sigma(\mathbf{h}_{i,t_i-1}\mathbf{\Gamma}), \tag{4}$$

where $\mathbf{\Gamma}$ is a learnable weight matrix parameter and σ is the sigmoid function. Using the correlation matrix, we get the following predictor vector for the i-th student of length m:

$$\mathbf{y}_i = \alpha(\mathbf{s}_i \odot \omega + \mathbf{s}_i * \mathbf{M}) + (1 - \alpha)\mathbf{s}_i, \tag{5}$$

where $\alpha \in [0, 1]$ is a learnable parameter used to control the balance between intra-basket correlative and inter-basket sequential associations of courses. The j-th element of $\mathbf{y}_i$ indicates the recommendation score of course c_j to be in the target basket of i-th student.

3.3 CourseDREAM

We propose the Course Dynamic Recurrent Basket Model (CourseDREAM), based on DREAM [33], to recommend a set of courses for the target semester. We consider two different latent representation techniques, max pooling and average pooling, to create a representation of a semester of courses. We use long short-term memory networks (LSTM) to capture the sequential transition of courses over the sequence of semesters and a dynamic representation of students which captures the dynamic interests of a student throughout their studies. For prediction, we calculate the score for each course $\forall p \in \mathcal{C}$ based on the most recent basket's $\mathcal{B}_{i,(t_i-1)}$ dynamic representation. We recommend the courses with the highest scores for the target semester.

Latent Representation of Semester Each basket of the i-th student consists of one or more courses. The j-th course that i-th student took at semester t has the latent representation $\mathbf{c}_{i,j,t}$ with d latent dimensions. We create a latent vector representation $\mathbf{r}_{i,t}$ for the set of courses that the i-th student took in semester t by aggregating the vector representations of these courses, $\mathbf{c}_{i,j,t}$. We use two types of aggregation operations. First, in max pooling, we take the maximum value of every dimension over these vectors. The l-th element ($l \in [1, d]$) of $\mathbf{r}_{i,t}$ is created as:

$$\mathbf{r}_{i,t,l} = max(\mathbf{c}_{i,1,t,l}, \mathbf{c}_{i,2,t,l}, ...), \tag{6}$$

where $\mathbf{c}_{i,j,t,l}$ is the l-th element of the course representation vector $\mathbf{c}_{i,j,t}$. Secondly, for the average pooling, we aggregate the courses' latent representations in semester t by taking the average value of every dimension, as follows:

$$\mathbf{r}_{i,t} = \frac{1}{|\mathcal{B}_{i,t}|} \sum_{j=1}^{|\mathcal{B}_{i,t}|} \mathbf{c}_{i,j,t}. \tag{7}$$

Next, these representations of the sequence of baskets are passed to the recurrent neural network (RNN) architecture.

Dynamic Representation of a Student We incorporate LSTM networks in the RNN architecture where the hidden layer $\mathbf{h}_{i,t}$ is the dynamic representation of i-th student at semester t. The recurrent connection weight matrix $\mathbf{W} \in \mathbb{R}^{d \times d}$ is helpful to propagate sequential signals between two adjacent hidden states $\mathbf{h}_{i,t-1}$ and $\mathbf{h}_{i,t}$. We have a learnable transition matrix $\mathbf{T} \in \mathbb{R}^{t_m \times d}$ between the latent representation of basket $\mathbf{r}_{i,t}$ and a student's interests, where t_m is the maximum length of the sequence of baskets for any student. We compute the vector representation of the hidden layer as follows:

$$\mathbf{h}_{i,t} = f(\mathbf{T}\mathbf{r}_{i,t} + \mathbf{W}\mathbf{h}_{i,t-1}), \tag{8}$$

where $\mathbf{h}_{i,t-1}$ is the dynamic representation of the previous semester. $f(\cdot)$ is the sigmoid activation function, i.e., $f(x) = 1/(1 + e^{-x})$.

Score generation The model generates a score $\mathbf{y}_{i,t_i}$ for all available courses that the i-th student might take at target semester t_i by using the matrix $\mathbf{M}$ with the latent representation of all courses and the dynamic representation of the student $\mathbf{h}_{i,t}$ as follows:

$$\mathbf{y}_{i,t_i} = \mathbf{M}^T\mathbf{h}_{i,t}, \tag{9}$$

where the j-th element of $\mathbf{y}_{i,t_i}$, represents the recommendation score for the j-th course.

Table 2: Data statistics

	# students	# courses	# target baskets
Training	2973	618	14070
Validation	1231	540	2743
Test	657	494	1259

4. EXPERIMENTAL EVALUATION

4.1 Dataset

We used a real-world dataset from Florida International University, a public US university, that spans 9 years. Our dataset consists of the course registration history of undergraduate students in the Computer Science department. The grades follow the A–F grading scale (A, A-, B+, B, B-, C+, C, D, F). We only consider the data of students who have successfully graduated with a degree. We remove instances in which a grade less than C was earned because these do not (usually) count towards degree requirements [17]. We also remove an instance if a student drops a class in the middle of the semester. In this way, we keep course sequences and information that at least lead to successful graduation and may be considered good examples. We remove courses that appear less than three times in our dataset.

After preprocessing, we have the course registration history of 3328 students and there are 647 unique courses. We split the data into train, validation, and test set. We use the last three semesters (summer 2021, fall 2021, and spring 2022) for testing purposes and the previous 3 semesters (summer 2020, fall 2020, and spring 2021) for validation and model selection. The rest of the data before the summer 2020 term (almost seven years' course registration history) are kept in the training set.

From the validation and test sets, we remove the courses that do not appear in the training set. We also remove any instances from the training, validation, and test set where the length of the basket sequence is less than three for a student. In other words, to generate recommendations we need the history of at least two previous semesters. The statistics of training, validation, and test data are presented in Table 2. Each student might correspond to multiple instances, one for each semester that could be considered as a target semester. For example, if a student took courses from fall 2019 until Spring 2021, they are considered in two instances on the test set (we generate recommendations from summer 2020 and spring 2021) and with three recommendations in the validation (target semesters: summer 2020, fall 2020, and spring 2021).

4.2 Evaluation metrics

As in prior work [13, 21, 23, 33], we used **Recall@k** score as our main evaluation metric, where k is the number of courses that the student took at the target semester.

$$\text{Recall@}k = \frac{\text{\# of relevant recommendations}}{\text{\# of actual courses in target semester}} \quad (10)$$

We essentially calculate the fraction of courses in the target semester that we correctly recommended to a student. In the subsequent sections, we report the average score over all the recommendations, i.e., target baskets. Recall and precision scores are equal as we recommend as many courses as the target student will take in the target semester. We also calculate the percentage of students for who we offer at least one relevant recommendation (**%rel**) as:

$$\%\text{rel} = \frac{\text{\# instances with} \geq 1 \text{ relevant recommendation}}{\text{\# total instances}} \quad (11)$$

This metric captures the ability of our recommendations to retrieve at least one relevant course for each student.

4.3 Experimental setting

We use the training set to build the models, and we select the parameters of the model with the best performance, based on the Recall@k metric, on the validation set. For the model selected, we calculate the evaluation metrics on the test set.

For the CourseBEACON model, for parameter α, we have tested the values [0.1, 0.3, 0.5, 0.7, 0.9]. α balances the importance of intra-basket correlation and inter-basket sequential association of courses. The lower value of α prioritizes the sequential association more than the intra-basket correlation; a higher value prioritizes the correlation of courses within the basket more. We also examined embedding dimensions=[16, 32, 64], hidden units=[32, 64, 128] of LSTM networks, and dropout rates=[0.3, 0.4]. For the CourseDREAM model, we used both max pooling and average pooling, however, the outcomes were very similar. In this paper, the results are reported with the average pooling technique. We tried LSTM layers=[1, 2, 3], embedding dimensions=[8, 16, 32], and dropout rates=[0.3, 0.4, 0.5, 0.6] for the CourseDREAM model.

4.4 Competing approaches

4.4.1 *Non-sequential baseline*

We use a popularity-based approach as a non-sequential baseline for course recommendation. Incorporating the hashing technique, we create a dictionary for each semester, starting from the first semester of each student, and count how many students take course $p \in \mathcal{C}$ in that semester of their studies. The top courses with the highest frequency at the t-th semester are recommended for the t-th semester of a target student. For example, if we have a student, and the target semester is his fourth semester in the program, we will recommend the most popular courses that students in the training set take in their fourth semester.

4.4.2 *Sequential baseline*

We use a popular sequence-based approach as our sequential baseline for course recommendation. For each pair of courses $\forall p, q \in \mathcal{C}$, we check how many times students took course q after course p. We create a bipartite graph with pairs of courses on consecutive semesters available in the training data. Courses are nodes and the weight of a connecting edge increases by 1 if one course comes before another course (i.e., weight(p, q) += 1 if course p is taken just before course q by a student). We normalize the weights as follows:

$$\text{weight}(p, q) = \frac{\text{weight}(p, q)}{\sum_{\forall r \in \mathcal{C}} \text{weight}(p, r)} \quad (12)$$

Given the courses that a target student took the previous semester, $\mathcal{B}_{i,t_i-1}$, we can recommend a set of courses for the t_i-th (target) semester. The recommendation score of a course is measured based on the summation of the weights from all the courses of the previous semester to this course in our created bipartite graph.

4.4.3 *Tensor decomposition*

We re-implement the session-based tensor decomposition technique [7] to recommend courses for the upcoming semester. This model prioritizes users' preferences of items in a basket over the sequential nature of items in the basket sequence of users. Using the training data, we create tensor $\mathbf{F} \in \mathbb{R}^{n \times m \times m}$, where $\mathbf{F}_{i,p,q}$ stores the number of times that courses p, q are taken together in the same semester from the i-th student. This tensor is very sparse, as there are many pairs of courses p, q that are not taken together by each student.

So, we use the RESCAL tensor decomposition technique with factorization rank, d, to get the approximate value of $\mathbf{F}_{i,p,q}$. We calculate the matrix $\mathbf{A}$ (course embedding, $\mathbf{A} \in \mathbb{R}^{m \times d}$) and tensor $\mathbf{R}$ (user embedding, $\mathbf{R} \in \mathbb{R}^{d \times d \times n}$), and then, we calculate $\tilde{\mathbf{F}}_{i,p,q} = \mathbf{Q}_p * \mathbf{A}_q^T$ where $\mathbf{Q}_p$ is the query vector, i.e., the dot product of $\mathbf{A}_p \in \mathbb{R}^{1 \times d}$ for course p and $\mathbf{R}_i \in \mathbb{R}^{d \times d \times 1}$ for i-th student. To speed up the recommendation process, we implement a hashing technique by using the approximate nearest neighbor (ANN) indexing library, ANNOY. In this case, the query vector, $\mathbf{Q}_p$, is calculated for any course p taken by i-th student and we find the courses q which are nearest neighbors to the query vector using annoy indexing and calculate $\tilde{\mathbf{F}}_{i,p,q}$ for those p, q pairs. Then, for the i-th student, we recommend the courses q that have the highest $\tilde{\mathbf{F}}_{i,p,q}$ scores based on the courses p that the student has already taken.

We tried different rank values for factorization, d=[1, 2, 3, 4, 5, 8, 10, 15, 20, 50], and different numbers of nearest neighbors [5, 40, 100] for ANN indexing. However, we observe better results when we do not use ANN indexing which takes the maximum number of nearest neighbors (all available courses) into consideration.

4.4.4 *Scholars walk*

We use the Scholars walk model [23], a non-session-based approach to recommend a set of courses for the next session. First, we calculate matrix $\mathbf{F} \in \mathbb{R}^{m \times m}$ that contains the frequency $\mathbf{F}_{p,q}$ of every pair of consecutive courses $p, q \in \mathcal{C}$. We normalize $\mathbf{F}$ to get the transition probability matrix, $\mathbf{T}$. Then we perform a random walk on the course-to-course graph that is described by $\mathbf{T}$. The probability of the walker reaching the vertices after K steps gives an intuitive measure that is useful to rank the courses for a student to offer a personalized recommendation. We use the scholars walk algorithm to perform a random walk with restarts which guides us to consider direct and transitive relations between the courses.

We tried the following value for the parameters: the number of steps allowed=[1,2,3,4,5]; *alpha*=[1e-4, 1e-3, 1e-2, 1e-1, 0.2, 0.4, 0.6, 0.7, 0.8, 0.85, 0.9, 0.99, 0.999]; *beta* values from 0 to 1.6 with step 0.1.

Table 3: Performance comparison in terms of Recall@k.

Model	Validation	Test
Non Sequential baseline	0.1600	0.1039
Sequential baseline	0.2991	0.2470
Tensor Decomposition	0.1596	0.1309
Scholars Walk	0.3619	0.2679
CourseBEACON	**0.3859**	0.2948
CourseDREAM	0.3856	**0.3023**

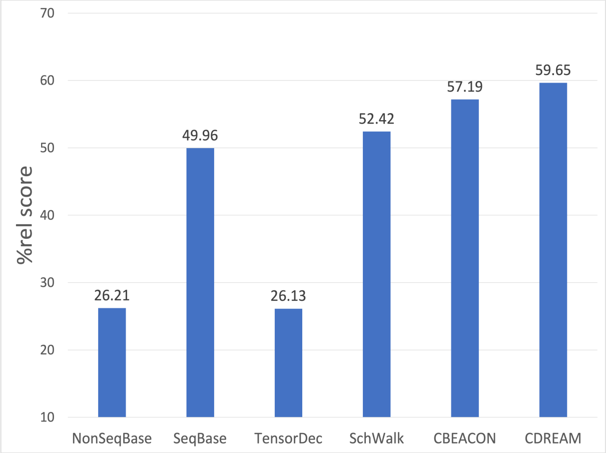


Figure 1: Percentage of instances with at least one relevant recommendation

4.5 Recommending courses

We recommend the top courses for the target semester $\mathcal{B}_{i,t_i}$ based on the predicted score $\mathbf{y}_{i,j}$ for course c_j for i-th student. The scores demonstrate how likely is for each course to be taken on the next semester with respect to both correlation of courses within the semester and sequential associations of courses over the semesters. We also create a list of courses for each semester t observing which courses are offered and available for all students. During post-processing, while recommending courses for a student, we filter out the courses which the student took in any previous semester and the courses which are not offered at that target semester [21, 23]. Then, we recommend the top $k = |\mathcal{B}_{i,t_i}|$ courses based on the highest scores for that student.

5. RESULTS

In this section, we will discuss the performance of our proposed approaches compared to the state-of-the-art sequential and non-sequential session-based or non-session-based approaches. We will also present how the hyperparameters affected the overall performance of our models.

5.1 Performance Comparison

The performance results of our proposed approaches and other competing approaches are shown in Table 3. We present Recall@k score for both the validation and test data. The percentage of relevant recommendations (%rel, percentage of at least one correct prediction for each instance) is presented in Figure 5.1.

First, if we only consider existing approaches, the best performing model is Scholars Walk. This model achieves recall performance around 26.79% and produces at least one relevant recommendation for 52.42% of the instances. On the opposite side, the non-sequential baseline performs particularly badly. The reason might be that in our school, we have a lot of transfer students, and the courses that they take in their second semester for example might be very different from traditional students.

Second, sequential approaches (sequential baseline, Scholars walk, CourseBEACON, CourseDREAM) outperform the non-sequential approaches (non-sequential baseline and tensor decomposition). This indicates that the sequence of courses over the semesters is an important factor in course selection. Sequential approaches can capture the student-course interaction along with the sequential transition of courses. On the contrary, non-sequential approaches just capture student-course interactions and students' preferences. We actually tested another non-sequential approach, BPR-MF, but the recall results were as low as the Tensor Decomposition, and we decided to only present one of these non-sequential approaches.

Third, our proposed session-based approaches outperform all the other competing approaches for course recommendation that we tested. We have the highest Recall@k and %rel with the CourseDREAM model. Between the two proposed models, CourseDREAM seems to be more stable, as there is a smaller difference between the validation and test performance. The fact that CourseDREAM behaves betters than CourseBEACON indicates that latent vector representation using the average pooling technique is more effective than creating the correlation matrix for the courses taken within a semester. An explanation could be that the correlation matrix may suffer from and be dominated by popular courses. We can also see that capturing the relationship of courses taken in a semester in the session-based approach is working better than other sequential approaches. The %rel scores of CourseDREAM demonstrate that we can recommend at least one correct recommendation for 59.65% of the instances.

Fourth, deep learning models (like LSTM networks) can capture the sequential transition of courses over the sequence of semesters. Incorporating the notion of the suitability of courses co-taken within a semester produces more accurate and useful recommendations.

5.2 The effect of different hyperparameters

First, we examine how the parameter α affects the results of the CourseBEACON model. In Fig. 5.2, we present the Recall@k of the validation and test set achieved for different values of α for the combination of parameters that have the best Recall@k (dropout rate = 0.4, embedding dimension d = 64, and hidden units = 128). We observe that lower values of α provide better performance, with the best model having $\alpha = 0.3$. This means that the sequential transition of courses plays more importance than the intra-basket correlation of courses within a semester. However, we still need to consider the relationship between courses taken in the same semester for course recommendation. This is what gives an advantage to these session-based models.

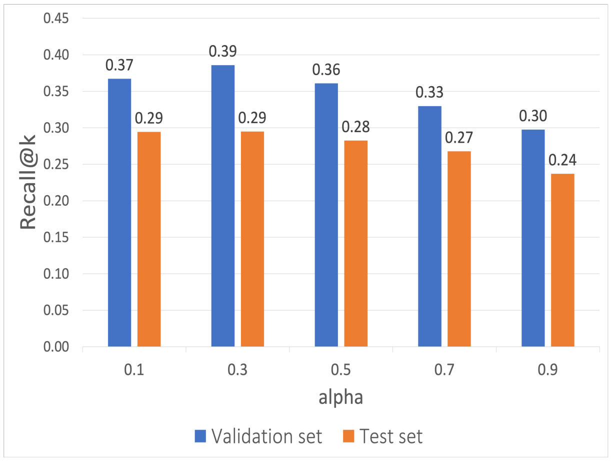


Figure 2: The effect of α in CourseBEACON model

Table 4: The effect of different hyperparameters in CourseDREAM model in terms of Recall@k (dropout rate = 0.4)

embedding dimensions	RNN layers	Validation	Test
32	1	0.3587	0.2782
32	2	0.3559	0.2792
32	3	**0.3856**	**0.3023**
16	1	0.3706	0.2882
16	2	0.3649	0.2943
16	3	0.3770	0.2990
8	1	0.3721	0.2958
8	2	0.3602	0.2826
8	3	0.3312	0.2659

Next, we examine the performance of the CourseDREAM model with respect to the parameters of embedding dimension and number of RNN layers in Table 4. Here, we have set dropout rates to 0.4, which gives us the best-performing model. We observe that the number of LSTM layers and the number of embedding dimensions do influence the results. Except for the case of 8 embedded dimensions, our models benefit from the increased number of RNN layers, which capture more complex patterns in the data.

6. CONCLUSION

We propose the use of session-based recommendation approaches for recommending suitable and complementary courses for the upcoming semester. In particular, we introduce CourseDREAM and CourseBEACON, two sequential session-based approaches that capture the relationship of the co-taken courses in different ways. Our experimental results show that our proposed models outperform all the sequential and non-sequential competing approaches. CourseDREAM can provide more relevant recommendations for the students so that recommended set of courses are related and more likely to be taken together within a semester. Our models will be helpful in advising students to achieve better performance overall and graduate on time.

7. REFERENCES

[1] A. Al-Badarenah and J. Alsakran. An automated recommender system for course selection. *International Journal of Advanced Computer Science and Applications*, 7(3):166–175, 2016.

[2] S. Boumi and A. E. Vela. Quantifying the impact of students' semester course load on their academic performance. In *2021 ASEE Virtual Annual Conference Content Access*, 2021.

[3] M. G. Brown, R. M. DeMonbrun, and S. D. Teasley. Conceptualizing co-enrollment: Accounting for student experiences across the curriculum. In *Proceedings of the 8th international conference on learning analytics and knowledge*, pages 305–309, 2018.

[4] S. Chockkalingam, R. Yu, and Z. A. Pardos. Which one's more work? predicting effective credit hours between courses. In *LAK21: 11th International Learning Analytics and Knowledge Conference*, pages 599–605, 2021.

[5] C. De Medio, C. Limongelli, F. Sciarrone, and M. Temperini. Moodlerec: A recommendation system for creating courses using the moodle e-learning platform. *Computers in Human Behavior*, 104:106168, 2020.

[6] A. Diamond, J. Roberts, T. Vorley, G. Birkin, J. Evans, J. Sheen, and T. Nathwani. Uk review of the provision of information about higher education: advisory study and literature review: report to the uk higher education funding bodies by cfe research. 2014.

[7] N. Entezari, E. E. Papalexakis, H. Wang, S. Rao, and S. K. Prasad. Tensor-based complementary product recommendation. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 409–415. IEEE, 2021.

[8] A. Esteban, A. Zafra, and C. Romero. A hybrid multi-criteria approach using a genetic algorithm for recommending courses to university students. *International educational data mining society*, 2018.

[9] J. Gardner and C. Brooks. Coenrollment networks and their relationship to grades in undergraduate education. In *Proceedings of the 8th international conference on learning analytics and knowledge*, pages 295–304, 2018.

[10] M. Hanson. *College Dropout Rates.* EducationData.org, June 17 2022. `https://educationdata.org/college-dropout-rates`.

[11] A. Kadlec, J. Immerwahr, and J. Gupta. Guided pathways to student success perspectives from indiana college students and advisors. *New York: Public Agenda*, 2014.

[12] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[13] D.-T. Le, H. W. Lauw, and Y. Fang. Correlation-sensitive next-basket recommendation. 2019.

[14] B. Ma, M. Lu, Y. Taniguchi, and S. Konomi. Exploration and explanation: An interactive course recommendation system for university environments. In *CEUR Workshop Proceedings*, volume 2903. CEUR-WS, 2021.

[15] B. Mondal, O. Patra, S. Mishra, and P. Patra. A course recommendation system based on grades. In *2020 international conference on computer science, engineering and applications (ICCSEA)*, pages 1–5. IEEE, 2020.

[16] S. Morsy and G. Karypis. Learning course sequencing for course recommendation. 2018.

[17] S. Morsy and G. Karypis. Will this course increase or decrease your gpa? towards grade-aware course recommendation. *arXiv preprint arXiv:1904.11798*, 2019.

[18] J. Naren, M. Z. Banu, and S. Lohavani. Recommendation system for students' course selection. In *Smart Systems and IoT: Innovations in Computing*, pages 825–834. Springer, 2020.

[19] R. Obeidat, R. Duwairi, and A. Al-Aiad. A collaborative recommendation system for online courses recommendations. In *2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*, pages 49–54. IEEE, 2019.

[20] A. Parameswaran, P. Venetis, and H. Garcia-Molina. Recommendation systems with complex constraints: A course recommendation perspective. *ACM Transactions on Information Systems (TOIS)*, 29(4):1–33, 2011.

[21] Z. A. Pardos, Z. Fan, and W. Jiang. Connectionist recommendation in the wild: on the utility and scrutability of neural networks for personalized course guidance. *User modeling and user-adapted interaction*, 29(2):487–525, 2019.

[22] Z. A. Pardos and W. Jiang. Designing for serendipity in a university course recommendation system. In *Proceedings of the tenth international conference on learning analytics & knowledge*, pages 350–359, 2020.

[23] A. Polyzou, A. N. Nikolakopoulos, and G. Karypis. Scholars walk: A markov chain framework for course recommendation. *International Educational Data Mining Society*, 2019.

[24] Z. Ren, X. Ning, A. S. Lan, and H. Rangwala. Grade prediction based on cumulative knowledge and co-taken courses. *International Educational Data Mining Society*, 2019.

[25] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[26] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 811–820, 2010.

[27] E. Shao, S. Guo, and Z. A. Pardos. Degree planning with plan-bert: Multi-semester recommendation using future courses of interest. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14920–14929, 2021.

[28] M. S. Sulaiman, A. A. Tamizi, M. R. Shamsudin, and A. Azmi. Course recommendation system using fuzzy logic approach. *Indonesian Journal of Electrical Engineering and Computer Science*, 17(1):365–371, 2020.

[29] P. Symeonidis and D. Malakoudis. Multi-modal

matrix factorization with side information for recommending massive open online courses. *Expert Systems with Applications*, 118:261–271, 2019.

[30] M. Wan, D. Wang, J. Liu, P. Bennett, and J. McAuley. Representing and recommending shopping baskets with complementarity, compatibility and loyalty. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1133–1142, 2018.

[31] S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. A. Orgun, and D. Lian. A survey on session-based recommender systems. *ACM Computing Surveys (CSUR)*, 54(7):1–38, 2021.

[32] C. Wong. Sequence based course recommender for personalized curriculum planning. In *International Conference on Artificial Intelligence in Education*, pages 531–534. Springer, 2018.

[33] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 729–732, 2016.

[34] H. Zhang, T. Huang, Z. Lv, S. Liu, and Z. Zhou. Mcrs: A course recommendation system for moocs. *Multimedia Tools and Applications*, 77(6):7051–7069, 2018.

In Search of Negative Moments: Multi-Modal Analysis of Teacher Negativity in Classroom Observation Videos

Zilin Dai
Worcester Polytechnic Institute
Worcester, MA, USA
zdai2@wpi.edu

Andrew McReynolds
Worcester Polytechnic Institute
Worcester, MA, USA
aamcreynolds@wpi.edu

Jacob Whitehill
Worcester Polytechnic Institute
Worcester, MA, USA
jrwhitehill@wpi.edu

ABSTRACT

We explore multi-modal machine learning-based approaches (facial expression recognition, auditory emotion recognition, and text sentiment analysis) to identify *negative moments* of teacher-student interaction during classroom teaching. Our analyses on a large (957 videos, each 20min) dataset of classroom observations suggest that: (1) Negative moments occur sparsely and are laborious to find by manually watching videos from start to finish. (2) Contemporary machine perception tools for emotion, speech, and text sentiment analysis show only limited ability to capture the diverse manifestations of classroom negativity in a fully automatic way. (3) Semi-automatic procedures that combine machine perception with human annotation may hold more promise for finding authentic moments of classroom negativity. Finally, (4) even short 10sec negative moments contain rich structure in terms of the actions and behaviors that they comprise.

Keywords

classroom observation analysis, multi-modal machine learning, speech analysis, sentiment analysis

1. INTRODUCTION

In school classrooms, the emotional climate set by the teacher can significantly impact student engagement, attitudes toward learning, and downstream academic and socioemotional outcomes [9, 8, 5]. Classrooms in which students feel encouraged, excited, and supported to learn are associated with positive engagement [9], fewer conflicts with teachers [16], and stronger executive functioning of the learners [28]. Conversely, classrooms with negative classroom climate – as exhibited by teacher irritability, anger, sarcasm, yelling, intimidation, etc. – are associated with poorer outcomes in these areas. Given the connection between classroom negativity and worse student outcomes, it is important to help teachers to reduce negativity in their teaching. Over the years, educational researchers have devised professional development and training programs to assist teachers in fostering classroom climates that are more conducive to learning [15]. One useful practice is to identify and discuss specific moments – either in the teacher's own classroom or in someone else's – that are especially positive or negative. For the positive moments, one can then examine the ways in which the teacher acted effectively; for the negative moments, one can discuss more constructive ways in which the teacher could have navigated the situation.

Needle in a Haystack: One obstacle to providing teachers with useful feedback on classroom observation is the need to find "teachable moments" that are worthy of close examination within a long classroom video. Even in a large library of classroom observation sessions, it may be difficult and laborious to find a variety of interesting moments. New methods for automated perception of school classrooms, as enabled by advances in computer vision, speech analysis, and natural language processing during the past 5-10 years, offer the possibility of accelerating the process of finding teachable moments. For an individual teacher, these new tools could make it possible to record their own teaching and quickly identify candidate moments – on a regular basis, not just 1-2 per year – that they should examine more closely. Deployed on a larger scale, such perceptual tools could also help researchers to systematically study moments of strong positivity or negativity in collections of classroom videos. In our paper, we assess the extent to which modern AI-based tools for the recognition of facial expression, auditory emotion, speech, and text sentiment could be used to find short (10sec) *negative moments* of classroom interaction between the teacher and the students.

Our definition of **negative moment** is rooted in the construct of *negative climate* from the Classroom Assessment Scoring System (CLASS; [25]). A classroom is said to exhibit negative climate if it contains negative affect (irritability, anger, harshness, *etc.*) by the teacher, punitive control, sarcasm/disrespect, or severe negativity (victimization, bullying, etc.). Negative climate under the CLASS framework is labeled on the timescale of 15-20 minute video segments. In contrast, we were interested in finding negative *moments* (10sec), as this is an arguably more useful timescale on which to give teachers *specific* feedback. This shorter timescale matches more closely with the specific actions and interactions that occur within a classroom teaching session (e.g., a single sentence spoken by the teacher to a student; physical actions such as touching or co-manipulation of an object by a teacher and student simultaneously; a facial expression that

Z. Dai, A. McReynolds, and J. Whitehill. In search of negative moments: Multi-modal analysis of teacher negativity in classroom observation videos. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 278–285, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115723

is displayed briefly for one person to another). It aligns with the natural timescale over which emotional states typically change [4]. We also are primarily interested in negativity expressed by the teacher, not by students.

Study Overview: We harness a large dataset of nearly 1000 videorecorded classroom observation sessions, each 20 minutes long, that were collected from individual teachers in elementary and middle schools. In terms of **research questions**, we examine (**RQ1**) to what extent modern AI-based machine perception tools can automatically find negative moments from classroom observation videos. In addition to fully automatic methods, we also explore (**RQ2**) whether a semi-automatic detection paradigm that combines AI with human annotation can yield a more accurate filtering mechanism. Finally, (**RQ3**) given the set of negative moments that we find, we explore what kind of semantic structure they contain and analyze them in terms of what happened on an utterance-by-utterance and action-by-action basis.

Ethics of Automated Classroom Analysis: Our long-term goal is to help teachers obtain more frequent and fine-grained feedback *about their own teaching* compared to the standard practice, which is to get very sparse feedback 1-2x/year from a school principal. Our paper provides a sober assessment of how realistic it is, using contemporary machine perception tools, to provide such feedback.

2. RELATED WORK

Classroom Observation Protocols: With the goal of characterizing classroom interactions more precisely and objectively, as well as providing teachers with more useful feedback, educational researchers have devised a variety of classroom observation protocols over the past two decades. These include the Protocol for Language Arts Teaching Observations (PLATO; [13]), Assessing Classroom Sociocultural Equity Scale (ACSES; [10]), and the Classroom Assessment Scoring System (CLASS; [25]). The CLASS is curriculum-agnostic and one of the most widely used protocols; it focuses on inter-personal *interactions* between teachers, students, and their peers.

Automatic Classroom Analysis: The EduSense system developed by Ahuja et al. [1] uses classroom audio and video to detect temporally specific features such as who is talking when, hand-raises, body posture, and smiles. These features can be aggregated over time and visualized in a dashboard for teachers that shows the total amount of instructor versus student speech, total number of hand raises, etc. The system does not perform high-level semantic analysis or make holistic judgments about the classroom experience. Zylich & Whitehill [29] trained custom neural networks to recognize key phrases associated with positive speech such as "please", "thank you", "good job", etc. They showed that the counts of these detected phrases over 15min classroom videos were correlated with some CLASS dimensions. Kelly et al. [19] developed a system to detect how often teachers are asking authentic questions of their students, *i.e.*, questions whose answers are open-ended and facilitate productive classroom discourse. Their approach takes an automatically generated transcript of the classroom audio; extracts word, sentence, and discourse-level features; and then applies regression trees to estimate the proportion, over the entire class period, of the teacher's questions that were open-ended. James et al. [18] used automatic facial expression recognition from classroom videos to estimate Positive and Negative Climate dimensions of the CLASS. Finally, Qiao and Beling [27] explored a multi-instance learning approach to identifying specific moments within classroom videos that human coders should examine in order to perform CLASS labeling more efficiently.

Figure 1: A random sample of 16 classroom videos (rendered at low resolution to preserve privacy) from our dataset.

3. DATASET

The dataset we used in our experiments (IRB #17-151 at Worcester Polytechnic Institute) was shared with our research group by a California-based company for teacher training. It consists of 957 classroom observation videos (20min each) ranging from kindergarten through middle school in a Midwestern state in the USA. Each video contains a different teacher and set of students. The videos were recorded by the teachers themselves to obtain feedback on their teaching; hence, the video camera model, placement, lighting, etc., can vary strongly between videos. While the teachers' faces and voices are usually clearly captured in each video, the students' often are not. See Figure 1.

4. MACHINE SENSORS

Our definition of negative moments involves the teacher's affect as well as the content of their speech and their actions. While capturing all facets of classroom negativity using automated tools is likely infeasible, there already exist machine perception tools that can detect certain aspects of negativity and that might help to find negative moments more quickly than by watching whole videos one-by-one. In particular, we explored the utility of modern (i.e., developed during the past 5 years) AI-based tools for speech recognition, text sentiment analysis, facial expression recognition, and auditory emotion recognition. We describe them below.[1]

4.1 Auditory Emotion Recognition

To analyze auditory emotion, we used the convolutional neural network described in [6]. The network takes a 162-dimensional feature vector (extracted by the Librosa package [21]) as input consisting of zero-crossing rates, Chroma-STFT, MFCC, RMS, and Mel spectrograms, which are all

[1] In addition to the individual sensors, we also tried an ensemble combining multiple sensors; however, the accuracy was no better than one of the individual sensors.

standard features in modern audio analysis. The features are extracted from 5sec audio segments, whereby each segment is split into multiple windows in time, and the features extracted from the windows are averaged before being passed to the network. The network was trained to classify 8 emotions (anger, calm, disgust, fear, happiness, neutral, sadness, and surprise) on a combination of 4 different datasets: CREMA-D [7], RAVDESS [20], SAVEE [17], and the TESS [26]. These datasets are widely used for auditory emotion recognition and contain recordings of individual adult speakers. They do not span the highly challenging conditions (overlapping speech, high background noise) found in school classrooms, nor do they contain children's voices; nonetheless, they are likely some of the best publicly available training datasets available. The test accuracy (61% over 8 emotions) of the trained network on these datasets is consistent with that reported by the authors of [6].

For our study, to obtain a speech-based emotion estimate for each 10sec moment of every classroom video, we split each moment into two 5-sec chunks, classified each chunk over the 8 emotion categories, and then averaged the estimates over the two chunks. Finally, to obtain an estimate of "negativity", we summed the emotion probabilities for the "anger" and "disgust" categories; since the focus of our study is on the *teacher*'s expressed negativity, we did not include the "sad" emotion in this sum. We note that, in practice, since most of the sound recorded in the videos comes from the teacher's speech, the auditory emotion detector is most likely to contain information on the teacher's expressed emotion rather than the students' auditory emotional responses.

Custom detectors: We also conducted a pilot experiment, using the same audio features, on training a custom detector (using 50 negative moments for training; see Section 5). The motivation was that training detectors on actual classroom data, rather than a general-purpose auditory emotion dataset, might be more effective. However, the test accuracy was basically at-chance, and we abandoned the approach.

4.2 Facial Expression Recognition

We first considered using OpenFace [3], but this software is specialized for analyzing a single face per image, not multiple faces and it detects facial Action Units [11] rather than semantic emotion labels ("anger", "disgust", etc.). Hence, we instead used the pre-trained facial emotion recognition convolutional neural network from [2], which achieves an overall accuracy, over a set of 7 detected emotions (anger, disgust, fear, happiness, sadness, surprise, neutral), of 66% on the FER2013 dataset [12]. FER2013 spans a wide range of lighting conditions and head poses (though not as extreme as those in classroom videos), but contains mostly adults.

To obtain a facial emotion estimate for each 10sec moment of every classroom video, we split each moment into 10 frames (spaced at 1 Hz); detected all the faces in the frame using OpenCV's built-in Haar-based cascaded face detector; and then analyzed the face for facial emotion using the trained emotion classifier. To compute an aggregate score for each emotion, we averaged the emotion estimates over all detected faces within the set of all 10 frames. (If no frames in the moment contained any detected faces, then the floating-point value NaN ("not a number") was assigned to all emotions in the 10sec moment.) Finally, to obtain a score of "negativity" for each moment, we added together the probability estimates for the "anger" and "disgust" emotions. We note that the facial expression sensor is most likely to contain information on the teacher's emotion, as the teacher's face is often the visual focus of the camera in most videos.

Summed over all sampled frames from all 957 classroom videos in the dataset, a total of 160398 faces were detected and analyzed for facial expression. On average, therefore, there were only about 0.14 faces detected per video frame, i.e., most people were not detected in most frames.

4.3 Text Sentiment Analysis

To analyze text for its sentiment, we first transcribed each video using the Web Speech API [22] developed by Mozilla and Google. Each video was split into 10sec chunks of audio, and each chunk was passed to the Web Speech API separately. The average number of 10sec moments in which the Web Speech API detected any speech at all was 80.19 (out of 120 total 10sec moments in a 20min video). The average number of transcribed words per video was 917.83. Each automatic transcription was then classified for sentiment using the Google Cloud Natural Language API. It returns a numeric score between -1.0 (most negative) and +1.0 (most positive) for each input. Examples: "a handle like it why do you think she got in his face and got upset with him" (sentiment: -0.9); "okay go ahead what's your favorite season" (sentiment: 0.4); and "very nice job on making your pros and cons very even very lined up makes it easy to count" (sentiment: 0.9). To obtain an estimate of "negativity" using the sentiment analyzer's raw output s, we remapped the range $[-1, 1]$ to $[0, 1]$ and reversed the scale, *i.e.*, the negativity n was computed as $n = 1 - (s/2 + 0.5) \in [0, 1]$.

5. FINDING NEGATIVE MOMENTS AUTOMATICALLY (RQ1)

In our first analysis we assess how accurately modern machine sensors can find classroom negative moments.

5.1 Annotation Process

Ideally, we would have ground-truth annotations of every 10sec moment of all 957 videos; however, this would be prohibitively expensive. Moreover, annotating a uniformly random sample from the dataset would likely uncover very few negative moments since they occur so sparsely. We thus use a different strategy: Since we have a form of automated labeling available to us (i.e., the sensors), we can use each sensor to find videos in which there is, according to the sensor's outputs, the largest *variance* of negativity. We then select the most negative and least negative moments (according to the sensors) within each of those videos, label these moments by hand, and then compute the accuracy of the machine w.r.t. human labels. With this procedure, we are essentially measuring the sensors' abilities to identify coarse-grained differences in negativity rather than very fine-grained differences if we had randomly selected pairs of moments from anywhere in the whole dataset. We applied this strategy for each of the three sensors as well as two ensemble models. All in all, we obtained 100 moments (20 from each automated method).

Table 1: Accuracy (AUC for absolute negativity, and proportion correct for relative negativity) of the different sensors used for fully automatic detection of classroom negative moments. Baseline for guessing is 0.5 in all cases.

Finding Negative Moments Automatically

Sensor	Absolute	Relative
Auditory Emotion	0.64	0.52
Facial Expression	0.41	0.35
Text Sentiment	0.61	0.52

The annotation team consisted of the three authors of this paper, of whom the senior author is CLASS-trained. Prior to annotation, the team examined a handful of video examples, and each annotator labeled them independently. Next, the team came together to discuss their labels and arrive at a consensus understanding. Finally, each labeler proceeded to annotate the remaining examples. We assessed inter-rater reliability (IRR) as the average pairwise agreement between annotators using the linearly weighted Cohen's κ coefficient.

5.2 Annotation Tasks

The labeling task consisted of both an *absolute* rating task and a *relative* rating task. The former is about distinguishing the negativity between any two moments of classroom teaching at any moment and from any teacher, whereas the latter is about comparing the negativity of two moments within the *same* teacher's classroom.

Absolute negativity: Annotators were presented with a set of 100 moments and were asked to rate each one as "negative", "positive", or "neutral". These labels were then converted into integers -1, 0, and +1, respectively. On this task, the average pairwise IRR was $\kappa = 0.39$. Over the 3×100 total labels across the three annotators, only 16 were negative. None of the 100 moments received a label of "negative" (-1) from all three labelers. Only 1 out of the 100 moments received 2 votes (out of 3) of "negative". These numbers reflect how classroom negativity often occurs very sparsely in a classroom observation session.

Relative negativity: Annotators were presented with a set of 50 *pairs* of 10sec moments, whereby each pair came from the same video but different pairs came from different videos. For each moment in each pair, they were asked to label which of the two moments was *more negative* (-1 if the first moment was more negative, and +1 if the second video was more negative), with an option for "neither" (0) if no difference in negativity could be discerned. On this task, the average pairwise IRR was $\kappa = 0.37$. Only 4 of the 50 moment-pairs received a unanimous vote across all 3 labelers that one moment was either "more negative" than the other.

5.3 Accuracy of Machine Sensors

Absolute negativity: To estimate each sensor's accuracy, we first averaged the three annotators' integer labels for each moment to obtain a "ground-truth" label. For instance, if two annotators labeled a moment as "neutral" and one labeled it as "negative", then the average is $-1/3$. We then computed binary labels for each moment (1 for "negative" and 0 for "non-negative") by thresholding this average with 0. After doing so, we obtained a set of 15 negative moments and 85 non-negative moments. We then computed the Area Under the ROC Curve (AUC) of each machine sensor using these binary labels. Using this procedure (see Table 1), we obtained an AUC of 0.64 for the auditory emotion sensor, 0.41 for the facial expression sensor (*i.e.*, slightly *worse* than just randomly guessing, though this is likely due to just statistical noise), and 0.61 for text sentiment.

Relative negativity: We selected the set of moment-pairs in which the average integer label (-1, 0, or +1) over the three annotators was non-zero, *i.e.*, the consensus was that one of the two moments in each pair was "more negative" than the other. This resulted in a set of 31 (out of the original) 50 moment-pairs. We then computed the fraction, for each machine sensor, of the pairs in which the sensor's output agreed with the average label. Using this procedure, we obtained a score (% correct) of 0.52 for the auditory emotion sensor, 0.35 for the facial expression sensor, and 0.52 for the text sentiment sensor. These accuracies are not significantly better than just randomly guessing (0.5 in this case).

5.4 Discussion

No sensor performed substantially above chance for either the absolute or relative negativity detection tasks, despite the fact that the data was sample was selected to have a high variance of negativity – i.e., the machine was tasked with discerning coarse-grained rather than fine-grained differences. Moreover, the IRR for both the absolute and the relative negativity labeling tasks was fairly low (0.3-0.4). This suggests that the machine sensors we tried had basically no ability to identify negative moments, and that randomly selecting moments from a video will uncover very few such moments of classroom interaction. This agrees with the annotation team's subjective experiences that there was little clear negativity in the moments they labeled.

Based on manually watching hundreds of classroom video segments, we suggest several possible explanations for why the sensors did not perform well: (1) The emotion categories recognized by the sensors do not closely match academic emotions [24] that occur in school classrooms. (2) The demographic diversity and difficulty of the training data is much more limited compared to the classroom videos in our dataset. (3) The face detector misses the majority of faces that occur in our video dataset; when it is visible, it is often difficult to perceive the person's facial expression.

With regards to the more promising results reported in [18, 29], we speculate that the larger timescale in their studies (15min) compared to ours (10sec) may help their models to "smooth out" measurement noise in the sensors' outputs.

6. FINDING NEGATIVE MOMENTS SEMI-AUTOMATICALLY (RQ2)

With the limited success of the fully automated approach, we next explored a *semi*-automatic approach that combines algorithmic filtering with human annotation. Our method was based on our observation that the automatic transcripts of the classroom videos, though imperfect, still hold insight into what transpired in each 10sec moment; moreover, in pilot data exploration we found that simple keyword searches for certain phrases such as "sit down" would already find moments in which the teacher was correcting students' behavior and possibly also exhibiting negativity. In particular, we heuristically formed a list of phrases that we deemed likely to contain moments of *behavioral corrections* [14], such as asking students to sit down, stop talking, pay attention, etc. Corrections are not inherently negative, particularly if the teacher redirects students toward more constructive behaviors and in a way that does not demean them. In practice, they are often associated with teacher negativity, and thus detecting behavioral corrections can help to uncover some (but by no means all) kinds of negative moments.

We assembled a list containing the following phrases that we deemed likely to capture situations that are associated with behavioral correction: "excuse me", "keep your", "why are you", "I need you", "stop", "be quiet", "sit down", "eyes on me", "can you please", "can you stop", "listen", "attention", "don't talk", "don't yell", "on your bottom"[2], "noise", and "keep the volume". We then devised the following procedure to identify "corrective" moments: (1) Use automatic speech recognition (ASR) to transcribe each 10sec moment from all the videos. (2) Filter the set of all moments to include all and only those that contain at least one of the keyphrases above. (3) Manually read the transcripts (but do not watch the corresponding video segment) of the filtered moments; keep only those that are deemed to be "corrective".

We performed the procedure above on our entire dataset of 957 classroom videos. In practice, we found the procedure to be both intuitive to perform – *i.e.*, the transcripts are usually quite readable and give some sense of the classroom interaction – and efficient – *i.e.*, it took only a few person-hours to read the transcripts filtered through step 2.

6.1 Annotation Process

To assess accuracy of the procedure, the annotation team examined 100 moments: 50 that passed step 3, and 50 that were filtered out during step 2 (since they did not contain any keyphrase). They labeled each moment as "negative" (-1), "neutral" (0), or "positive" (+1). To do so, they examined these 10sec moments *with* the video (*i.e.*, not just from the transcript like in step 3), including a few seconds of context before/after the start/end of each video segment so as to understand the moment more thoroughly. The average pairwise IRR on this task was $\kappa = 0.60$. In a similar manner, the team also labeled each moment as "corrective" vs. "not corrective" (IRR: $\kappa = 0.8$).

6.2 Accuracy of Semi-Automatic Procedure

Negative moments: Of the 50 moments that passed step 3 of the semi-automatic procedure, 29 (*i.e.*, 58%) were confirmed – by taking the average numeric label across all 3 labelers and thresholding at 0 – to be "negative". Of these 29 moments, 26 were further confirmed as "corrective". Moreover, there were 12 moments in which all 3 labelers unanimously agreed were negative, and 5 more moments in which 2 out of 3 labelers agreed were negative. The AUC of the semi-automatic procedure for distinguishing between negative and non-negative moments was 83.3%.

Corrective moments: Of the 50 moments that passed step 3 of the semi-automatic procedure, 33 (*i.e.*, 66%) were confirmed by the labelers, after taking majority vote of their corrective vs. not corrective labels, as being corrective. The AUC of the procedure for distinguishing between corrective and non-corrective moments is also 83.3%.

6.3 Discussion

This semi-automated procedure showed more promise for accurately finding negative moments than did the fully automated sensors. The IRR of manually validating the output of the procedure was also much higher (0.6 compared to 0.3-0.4 for labeling the results of the fully automated approach) and provides further validation that it is making meaningful distinctions in negativity.

When examining the false detections – *i.e.*, moments output by the procedure that were not actually negative – we found several in which the teacher was talking *about* negativity (*e.g.*, about why it is important to follow rules in society), rather than actually *exhibiting* negativity. This semantic distinction would likely be very difficult for a machine to make automatically. Another source of false detections that we found was the transcription error made by the Web Speech API, such that a keyphrase in our list was not actually spoken within the video. In terms of missed detections – *i.e.*, negative moments that were missed by the procedure – there are likely many kinds of classroom negativity that are not associated with corrective behavior and would thus be missed. However, by assembling a different list of keyphrases and/or applying more sophisticated methods of analyzing the transcripts, it is possible that other kinds of negative moments could also be discovered.

7. MANIFESTATIONS OF NEGATIVITY

Given that the machine sensors showed little success in uncovering negative moments, we wanted to examine whether this was because the negative moments in our dataset truly do not actually exhibit any differences in facial expression and/or auditory emotion, or whether the detectors we used were too poor in accuracy or perhaps not trained on the right kinds of data. To this end, we performed further annotation about which of the two 10sec moments in a pair from the same video are "less negative" (-1) or "more negative" (+1) in terms of *facial expression*, and (separately) in terms of *auditory emotion*. If no difference could be ascertained, a label of 0 was assigned. Importantly, the focus of this annotation task was to examine the facial and auditory emotion in isolation, and to ignore higher-level semantics of the content of the teacher's speech or the trajectory of their actions. We performed the annotation on the same set of 100 videos described in Section 6.1.

7.1 Negative Auditory Emotion

When judging which of the two moments exhibited more negative *auditory emotion*, the average pair-wise IRR of the annotators was $\kappa = 0.32$, suggesting low to moderate agreement on individual moments. This number agrees with our subjective impression that discerning differences in negativity based on auditory emotions is challenging, and that the differences are much smaller than, say, the difference between "happy" and "angry" in standard datasets used for

[2] a phrase sometimes told to young students to sit down

training speech emotion classifiers (Section 4.1). Nevertheless, once we *averaged* all three labelers' responses for each moment-pair, we found stronger evidence that the auditory emotion of a moment is diagnostic for labeling it as "negative": in 78% of the moment-pairs, the moment that was identified as having "more negative" *audio* was the moment in the pair that was labeled as a "negative moment" overall.

7.2 Negative Facial Expression

When examining facial expression, the IRR was 0.40, which was slightly higher than for auditory emotion. After taking the average label across all three annotators, we found that, in only 58% of the moment-pairs was the moment identified as having "more negative" *facial expression* the moment in the pair that was labeled as a "negative moment" overall.

7.3 Discussion

Together, these results suggest that, while there is some relationship between the facial and/or auditory emotion of the classroom and the overall negativity of each 10-second classroom moment, there is still considerable subjectivity when judging each individual moment. Similar to our results on fully automated approaches to finding negative moments with different sensors, here too we found that auditory emotion was more informative than facial expression. All in all, it seems that examining auditory and facial expressions in isolation is insufficient – what defines classroom negativity depends on more detailed analysis of what transpires.

8. NEGATIVE MOMENT ANALYSIS (RQ3)

To understand better the *semantic structure* of negative moments, we examined a set of 43 video clips that were labeled by our annotation team as "negative moments" in our previous analyses on fully automatic (Section 5) as well as semi-automatic (Section 6) methods for finding classroom negativity. We qualitatively examined each video clip to obtain a deeper understanding of the *subject* (the nature or cause) of the negativity, as well as the *trajectory* of actions and utterances that the 10sec moment comprised. As an example, the subject of several negative moments was the teacher asking students to sit down in their seats. This might involve actions and utterances such as pointing to the student's seat, approaching the student's desk, and directing the student to sit down. Through our qualitative coding process (described below), we identified 4 recurrent subjects: "Stop Fidgeting", "Sit Down", "Listen", and "Stop Talking". Further, we identified 6 types of actions & utterances: Direct Correction (expressed either verbally or physically) of the student's behavior, Sarcasm, Threat, Body Motion (e.g., aggressive posturing of the teacher's body w.r.t. the student), Deflection (e.g., brushing off a student's comment through a verbal rejoinder), and Justification (explaining why the teacher is correcting the student's behavior). See Table 2.

Procedures: The review process of the negative moments went as follows: *(1)* The annotation team watched the moment two times in a row together to gain a preliminary understanding; *(2)* The annotators discussed their opinions of the moments, how they believed each moment to break down into multiple stages, and what they believed the trajectory of actions and utterances to be; *(3)* The annotators watched the moment, pausing at notable points in time, to agree or disagree on each other's labels; and finally, *(4)* the annotators formed a consensus on the label trajectory of actions/utterances in the 10sec moment. The qualitative codes we used to analyze each clip, along with illustrative examples, can be found in Table 2.

Results: Through the analysis of the 43 10-second video clips we categorized using Table 2, we found that teachers, on average, performed about 2 actions ($\overline{X} = 2.09$, $SD = 1.00$) per 10sec moment. Some negative moments even contained up to 4 distinct actions/utterances. The action frequencies can be seen in Table 3, where each column corresponds to a different stage with each moment's trajectory.

8.1 Vignettes of Classroom Negativity

To give a more vivid sense of what kinds of negative moments emerged, we describe three "vignettes" that illustrate different *subjects* of negativity that we identified.

8.1.1 Vignette #1: Stop Fidgeting

*There is a small round table in the classroom with four students (likely between grades 2 and 4) surrounding it, with a teacher standing a few feet away. The teacher is standing next to a whiteboard with math (*i.e. $4 \times 5 = 20$*) written down. The teacher is providing instructions to the group of students on how to complete a printed assignment in front of each student. Most students are sitting still, watching the teacher, and looking at their papers. However, one student, who appears to be African-American, who is closest to the camera, and whose back is facing the camera, begins to dance in her seat: Her left arm is angled down towards the floor, and her right arm is angled up towards the ceiling; she is rocking her shoulders forward and back, causing her arms to sway. The teacher is distracted by the dancing, looks at the student with an angry expression, and then issues a verbal command with a harsh tone:* ***"I need you to stop. Thank you."*** *[Direct Correction – Verbal]. The teacher then turns to look at a boy seated at the table, who says something to the teacher which elicits a verbal response of* ***"Oh great, great"****.*

We speculate that this student's body movements and expressiveness might be an instance of verve, which is a learning style associated with African-American students that "can be defined as having energy, being intense, having expressive body language, and having a tendency to attend to several different areas of focus"; it is sometimes misinterpreted by teachers as challenging or assertive [14]. The last comment ("Oh great, great") was spoken in a tone that sounded sarcastic. This is a case where accurate and temporally precise recognition of negative auditory emotion is important to correctly interpret a teacher's action.

8.1.2 Vignette #2: Sit Down

About 15 students (between grades 1 and 3) are sitting on a large carpet with the teacher sitting on a rocking chair in front of the students. The moment begins with the teacher speaking to one male near the back of the carpet, asking him to sit down [Direct Correction – Verbal]. Her voice becomes more stern when she realizes multiple students are not following the direction to sit down. Her facial expression becomes more frustrated, and she states, ***"If I have to remind the boys in the back how to sit sharp one more***

Table 2: Types of Actions & Utterances within Negative Moments

Teacher Action	Descriptive Example
Direct Correction (Verbal)	*The teacher is counting down from five to have her class be quiet. When she reaches zero, she says, "shhhhhh", to have the last few students be quiet.*
Direct Correction (Physical)	*The teacher verbally tells the child, "No, no", while physically gesturing with her hand for the child to direct them to stop talking.*
Sarcasm	*"[Name], we will hear from you first... since you are eager to speak."*
Threat	*"I want you to put this stuff away and follow directions, or I am going to have to call dad... and grandma again, ok?"*
Body Motion	*The teacher is providing instructions to the class and the child in front of her is playing with a plastic bag, which leads to the teacher physically removing it from the child's hands.*
Deflection	*A child walks to the front of the room when they aren't supposed to. The teacher walks them back to their seat. The child protests; the teacher replies, "Ok, I am not hearing any of that."*
Justification	*"I see a lot of people who are off task... so we need to bring our attention up front."*

Table 3: Frequency of Actions Types in Negative Moments

	First	**Second**	**Third**	**Fourth**	**Total**
Direct Correction					
Verbal	15	13	5	–	33
Physical	12	8	3	2	25
Sarcasm	2	3	1	1	7
Threat	2	–	2	1	5
Body Motion	8	–	–	–	8
Deflection	1	–	–	1	2
Justification	3	5	2	–	10

time, you are going to lose points" *[Threat] while giving a single downward nod followed by her pointing behind the easel. At this point, the students sit properly and the teacher, after waiting a few seconds, resumes teaching.*

8.1.3 Vignette #3: Listen

The classroom consists of about 15 students (likely between grades 4 and 6) all situated at large communal tables in groups of two to four, and one teacher. The teacher is walking to the front of the room while discussing Día de Los Muertos (Day of the Dead) when she looks up and notices a group of students in the back of the class who are not on task. The teacher stops walking around the room, looks at the boys and, with a serious facial expression, she says: ***"Boys? I hope you are listening, don't play with the folder..."*** *[Direct Correction – Verbal]. While making her comments to the boy, she extends her arm [Direct Correction – Physical] and motioning for them to stop. After a short pause to make sure the boys are listening, the teacher resumes teaching.*

8.2 Discussion

Within the moments we analyzed, the Direct Correction action was most frequent. Most moments contained multiple distinct actions within them, despite the short duration (10sec). One of the least frequent actions we observed was Justification, even though this would likely be beneficial to students. Finally, in order to fully understand what happened as well as the intensity of each negative moment, the annotation team found it was necessary to combine information about *what* was said or done (semantic content), *how* it was said (tone of voice, facial expression), and what gestures and body language *accompanied* the action/utterance. The particular facial expressions and body movements that we observed in the vignettes were often short (<1 sec), which makes automatic detection even more challenging.

9. CONCLUSIONS

We conducted a machine learning analysis of how different automated tools for facial expression recognition, auditory emotion recognition, speech recognition, and text sentiment analysis can be used to identify classroom "negative moments" automatically. We considered both fully automatic as well as semi-automatic (*i.e.*, speech recognition combined with some human annotation) approaches to finding negative moments in a large collection (957 videos, 20min long) of classroom videos. Moreover, we examined, on an utterance-by-utterance and action-by-action level, a set of 43 negative moments that were found by the semi-automated procedure.

Lessons learned: (1) Negative moments occur rarely, and a random sample from a classroom observation is unlikely to contain many of them. (2) The differences in facial and auditory emotion that distinguish negative moments from normal instruction are subtle – much more so than the differences in emotion categories (happy, sad, etc.) found in contemporary emotion datasets. (3) Full automation of the search process for negative moments is very challenging for contemporary AI systems that are trained on basic emotions such as happy, sad, angry, etc. We found more promise in a simple semi-automated procedure that combines automatic speech recognition, keyphrase search, and some human annotation. (4) Even short 10sec negative moments often comprise multiple actions and/or utterances by the teacher.

Future research can explore whether large language models (LLMs) such as ChatGPT [23] can be trained (by fine-tuning and/or few-shot learning) to identify classroom negativity more accurately. One bottleneck, however, is the accuracy of speech recognition, especially given the noisy classroom conditions with overlapping and sometimes inaudible speech. In addition, training custom multimodal detectors of new behaviors and states such as "fidgeting", "sarcasm", etc., could be useful to understand classroom interactions.

Acknowledgment: This research was supported by the NSF National AI Institute for Student-AI Teaming (iSAT) under grant DRL #2019805. The opinions expressed are those of the authors and do not represent views of the NSF. We also gratefully acknowledge NSF awards #2046505 and #1822768.

10. REFERENCES

[1] K. Ahuja, D. Kim, F. Xhakaj, V. Varga, A. Xie, S. Zhang, J. E. Townsend, C. Harrison, A. Ogan, and Y. Agarwal. Edusense: Practical classroom sensing at scale. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(3):1–26, 2019.

[2] O. Arriaga, M. Valdenegro-Toro, and P. Plöger. Real-time convolutional neural networks for emotion and gender classification. *arXiv preprint arXiv:1710.07557*, 2017.

[3] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency. Openface 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 59–66. IEEE, 2018.

[4] Y. Baveye, E. Dellandrea, C. Chamaret, and L. Chen. Liris-accede: A video database for affective content analysis. *IEEE Transactions on Affective Computing*, 6(1):43–55, 2015.

[5] M. Burchinal, L. Vernon-Feagans, V. Vitiello, M. Greenberg, F. L. P. K. Investigators, et al. Thresholds in the association between child care quality and child outcomes in rural preschool children. *Early childhood research quarterly*, 29(1):41–51, 2014.

[6] S. Burnwal. Speech emotion recognition, 2020.

[7] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma. Crema-d: Crowd-sourced emotional multimodal actors dataset. *IEEE transactions on affective computing*, 5(4):377–390, 2014.

[8] T. W. Curby, L. L. Brock, and B. K. Hamre. Teachers' emotional support consistency predicts children's achievement gains and social skills. *Early Education & Development*, 24(3):292–309, 2013.

[9] T. W. Curby, J. T. Downer, and L. M. Booren. Behavioral exchanges between teachers and children over the course of a typical preschool day: Testing bidirectional associations. *Early Childhood Research Quarterly*, 29(2):193–204, 2014.

[10] S. M. Curenton, I. U. Iruka, M. Humphries, B. Jensen, T. Durden, S. E. Rochester, J. Sims, J. V. Whittaker, and M. B. Kinzie. Validity for the assessing classroom sociocultural equity scale (acses) in early childhood classrooms. *Early Education and Development*, 31(2):284–303, 2020.

[11] P. Ekman and W. V. Friesen. Facial action coding system. *Environmental Psychology & Nonverbal Behavior*, 1978.

[12] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, et al. Challenges in representation learning: A report on three machine learning contests. In *International conference on neural information processing*, pages 117–124. Springer, 2013.

[13] P. Grossman. Protocol for language arts teaching observations, 2009.

[14] M.-B. Hamilton and L. DeThorne. Volume and verve: Understanding correction/behavioral warnings in teacher–child classroom interactions involving an african american kindergarten student. *Language, Speech, and Hearing Services in Schools*, 52(1):64–83, 2021.

[15] B. Hamre, J. T. Downer, F. M. Jamil, and R. C. Pianta. Enhancing teachers' intentional use of effective interactions with children: Designing and testing professional development interventions. *Handbook of early childhood education*, pages 507–532, 2012.

[16] B. K. Hamre, R. C. Pianta, J. T. Downer, and A. J. Mashburn. Teachers' perceptions of conflict with young students: Looking beyond problem behaviors. *Social Development*, 17(1):115–136, 2008.

[17] P. Jackson and S. Haq. Surrey audio-visual expressed emotion (savee) database. *University of Surrey: Guildford, UK*, 2014.

[18] A. James, M. Kashyap, Y. H. V. Chua, T. Maszczyk, A. M. Núñez, R. Bull, and J. Dauwels. Inferring the climate in classrooms from audio and video recordings: a machine learning approach. In *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, pages 983–988. IEEE, 2018.

[19] S. Kelly, A. M. Olney, P. Donnelly, M. Nystrand, and S. K. D'Mello. Automatically measuring question authenticity in real-world classrooms. *Educational Researcher*, 47(7):451–464, 2018.

[20] S. R. Livingstone and F. A. Russo. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391, 2018.

[21] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, pages 18–25. Citeseer, 2015.

[22] A. Natal, G. Shires, and P. Jägenstedt. Web speech api draft community group report, 2020.

[23] OpenAI. Gpt-4 technical report, 2023.

[24] R. Pekrun and L. Linnenbrink-Garcia. Academic emotions and student engagement. In *Handbook of research on student engagement*. Springer, 2012.

[25] R. C. Pianta, K. M. La Paro, and B. K. Hamre. *Classroom Assessment Scoring System™: Manual K-3*. Paul H Brookes Publishing, 2008.

[26] M. K. Pichora-Fuller and K. Dupuis. Toronto emotional speech set (tess), 2020.

[27] Q. Qiao and P. A. Beling. Classroom video assessment and retrieval via multiple instance learning. In *International Conference on Artificial Intelligence in Education*, pages 272–279. Springer, 2011.

[28] C. Weiland, K. Ulvestad, J. Sachs, and H. Yoshikawa. Associations between classroom quality and children's vocabulary and executive function skills in an urban public prekindergarten program. *Early Childhood Research Quarterly*, 28(2):199–209, 2013.

[29] B. Zylich and J. Whitehill. Noise-robust key-phrase detectors for automated classroom feedback. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 9215–9219. IEEE, 2020.

Can’t Inflate Data? Let the Models Unite and Vote: Data-agnostic Method to Avoid Overfit with Small Data

Machi Shimmei
North Carolina State University
mshimme@ncsu.edu

Noboru Matsuda
North Carolina State University
Noboru.Matuda@ncsu.edu

ABSTRACT

We propose an innovative, effective, and data-agnostic method to train a deep-neural network model with an extremely small training dataset, called VELR (Voting-based Ensemble Learning with Rejection). In educational research and practice, providing valid labels for a sufficient amount of data to be used for supervised learning can be very costly and often impractical. The shortage of training data often results in deep neural networks being overfitting. There are many methods to avoid overfitting such as data augmentation and regularization. Though, data augmentation is considerably data dependent and does not usually work well for natural language processing tasks. Moreover, regularization is often quite task specific and costly. To address this issue, we propose an ensemble of overfitting models with uncertainty-based rejection. We hypothesize that misclassification can be identified by estimating the distribution of the class-posterior probability P(y|$\boldsymbol{x}$) as a random variable. The proposed VELR method is data independent, and it does not require changes to the model structure or the re-training of the model. Empirical studies demonstrated that VELR achieved classification accuracy of 0.7 with only 200 samples per class on the CIFAR-10 dataset, but 75% of input samples were rejected. VELR was also applied to a question generation task using a BERT language model with only 350 training data points, which resulted in generating questions that are indistinguishable from human-generated questions. The paper concludes that VELR has potential applications to a broad range of real-world problems where misclassification is very costly, which is quite common in the educational domain.

Keywords

Ensemble learning with rejection, natural language processing, deep neural network, extremely low data regime, overfit.

1. INTRODUCTION

When applying a deep-neural network to real-world classification tasks, it is sometimes the case that only a very small amount of labeled data is available for training a model. When a deep neural-network (DNN) model is trained with a small amount of data, the model often overfits to the training data due to over-parameterization. We call such a problematically small amount of data the *extremely low data regime* [36].

Regularization is a widely used technique to prevent the model from overfitting. However, it requires the hyperparameters to be fine-tuned a priori, and the model must be retrained each time the hyperparameters are changed.

Another commonly used technique that is known to be an effective solution to the overfitting problem is semi-supervised learning, which utilizes unlabeled data in conjunction with labeled data for training [30, 33]. In recent years, data augmentation using Generative Adversarial Networks (GAN) has been actively studied to synthetically inflate data, significantly improving the performance of semi-supervised learning [4, 6, 10, 19]. However, there are situations where only a small amount of labeled data is available *and* data augmentation is not a suitable option. Text analysis in natural language processing is an example of one such data-augmentation incompatible task.

Although some research has demonstrated that DNN models can generalize well with extremely small data regimes, the performance is still lower than that of when an abundant amount of data is available [26, 32]. Low performance due to overfitting is a serious problem, especially when the model is used for real-world tasks where misclassification can be very costly and even unethical such as medical diagnoses or educational interventions. To further expand the application of DNN to real-word tasks, it is therefore critical to develop a technique that can overcome the overfitting problem with extremely low data regimes.

In this study, we propose a rigorous ensemble technique for estimating class-posterior probabilities based on *a collection of overfitting models*. Our proposed method does not use any regularization techniques or generative models for data augmentation to avoid overfitting. Instead of *preventing* overfitting while training models, we propose to identify unreliable classification using a soft voting ensemble method *based on the distribution of the estimated class-posterior probability* $P(y|\boldsymbol{x})$ among the collection of overfitting models.

In other words, we aggregate the class-posterior probabilities $P(y|\boldsymbol{x})$ from multiple isomorphic models (aka soft voting) instead of aggregating the class prediction y (aka hard voting) [37]. We treat $P(y|\boldsymbol{x})$ as a random variable while considering a predicted class-posterior probability from each model as an observation to estimate the distribution of this random variable.

An unreliable classification will be rejected to reduce the risk of giving wrong predictions. We shall call our proposed method ***V**oting-based **E**nsemble **L**earning with **R**ejection* (VELR).

With a lack of theoretical work in the design of a voting technique, we explored two soft-voting methods: min-majority voting and uniform voting. The min-majority voting estimates Gaussian Mixture Models and takes the minimum probability in a majority

M. Shimmei and N. Matsuda. Can’t inflate data? let the models unite and vote: Data-agnostic method to avoid overfit with small data. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 286–295, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115705

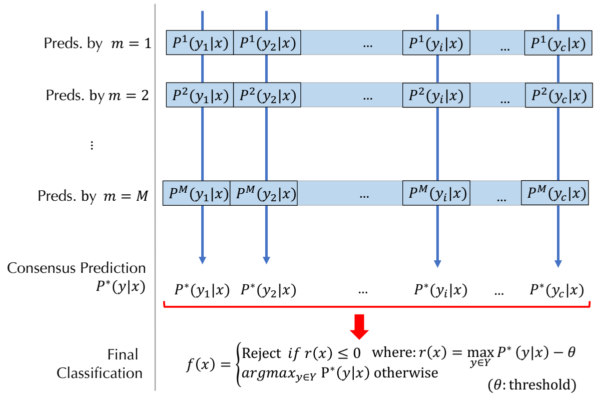


Figure 1. Set of posterior probability (or "certainty") $P^m(y_i \in Y|x)$ computed by a collection of models.

cluster, whereas uniform voting sums the probabilities with a uniform weight. Although uniform voting itself is not novel, *voting among overfitting models due to the extremely low data regime* has not been studied, as far as we are aware.

In addition, it is not clear in the current literature how classification with rejection works in conjunction with voting over an ensemble of overfitting models. We demonstrated that classification with rejection with voting shows a better performance than that with a single model when only an extremely low data regime is available.

To validate VELR, we conducted evaluation studies on two tasks: (1) image classification on a commonly used bench-mark dataset and (2) pedagogical question generation for online courseware engineering. The results showed that voting-based ensemble learning with rejection was able to identify incorrect predictions and accuracy of classification increased significantly by rejecting those predictions.

Our contributions are as follows: (1) We propose voting-based ensemble learning with rejection, VELR, a practical and data-agnostic solution for training deep-neural network models with extremely small datasets that would otherwise be overfit to the training data. (2) We show that a combination of soft voting among overfitting models and rejection can significantly increase performance of a model that relies on estimation of a class-posterior probability. (3) We demonstrated that VELR is data agonistic through two empirical studies—image and text analyses. (4) The code and data used for the current study have been open sourced[1].

2. VELR: VOTING-BASED ENSEMBLE LEARNING WITH REJECTION

2.1 Training the Base Models

VELR applies to any deep-neural network model that outputs normalized posterior probability (or *certainty*), $P(y|\boldsymbol{x}) = [0, 1]$, which means that when multiple certainties are output (e.g., multi-label classification), the sum of $P(y_i|\boldsymbol{x})$ are 1 across all outputs. In the current paper, we assume multiple certainties are output, but it sshould be clear that the same logic applies to models with a single certainty, e.g., a binary classification.

Suppose we have an input $\boldsymbol{x} \in \mathrm{X}$ in a multi-dimensional space and class labels $Y = \{y_1, y_2, \ldots, y_C\}$. In general, to train a classification model is to optimize a set of certainties $P(y_i \in \mathrm{Y}|\boldsymbol{x})$ in a training dataset.

When trained with an extremely low data regime, the model will unavoidably overfit. We therefore propose to create a collection of models that are independently trained using the same deep-neural network structure, the same training dataset, and the same hyperparameter settings. It is only that the random initial weights are different. Accordingly, a set of certainty $P^m(y_i \in \mathrm{Y}|\boldsymbol{x})$ for a sample $\boldsymbol{x}$ are computed, each independently by an individual model m ($m = 1, \ldots, \mathrm{M}$) as depicted in Figure 1. The question is how to make a consensus among multiple certainties. The next section describes a voting technique to compute the consensus certainty $P^*(y_i \in \mathrm{Y}|\boldsymbol{x})$.

2.2 Voting on Estimated Certainty Distribution

An essential problem of ensemble learning is to determine which posterior probability, among a collection of competing ones, should be taken. In the current literature, one approach takes model as the unit of analysis—i.e., individual models make a prediction based on their own posterior probabilities and then a majority vote is taken from the set of those predictions, aka hard voting [2].

VELR takes a different approach, where certainty is used as the unit of analysis. Namely, for each class $y_i \in Y$, VELR makes an ensemble decision about the posterior probability $P^*(y_i \in Y|\boldsymbol{x})$ based on a set of certainties, $P^m(y_i \in Y|\boldsymbol{x})$, $m = 1, \ldots, \mathrm{M}$, as shown in Figure 1. In the current literature, this approach is called soft voting [37]. In the rest of this paper, we call $P^*(y_i \in Y|\boldsymbol{x})$ as the *consensus certainty*[2].

We explored two different methods for voting: min-majority voting and uniform voting, as shown in the following subsections. Our basic hypothesis is that voting decisions should be made based on the distribution of the certainty $P(y_i|\boldsymbol{x})$ per class y_i among the M models. Therefore, we define a random variable $v^{y_i} = \{v_{x,m}^{y_i} = P^m(y_i|\boldsymbol{x});\ m = 1, \ldots, M\}$ for each sample $\boldsymbol{x}$ and class y_i. We hypothesize that the decision of classification should be made based on voting among v's.

2.2.1 *Min-majority voting*

For the min-majority voting, we assume that v^{y_i} follows the Gaussian Mixture Model (GMM) defined as:

$$P(v^{y_i}) = \sum\nolimits_{k=1}^{K} \pi_k \mathcal{N}(v^{y_i}|\mu_k, \sigma_k)$$

$$\Sigma_{k=1}^{K} \pi_k = 1,$$

$\mathcal{N}(v^{y_i}|\mu_k, \sigma_k)$: Gaussian Density function
K: Number of clusters

As Salman and Liu [25] analyzed, when models are overfitting, the probability distribution of the random variable v tends to skew towards 0 and 1. We therefore assume $K = 2$ in the current implementation of VELR.

For each sample $\boldsymbol{x}$, the estimation of π, μ, and σ is done by the EM algorithm [7] over the random variable v as mentioned above.

[1] The code and data are available at https://github.com/IEClab-NCSU/VELR

[2] We use the term "posterior probability", "prediction", and "certainty" interchangeably unless otherwise noted.

Once the density functions are estimated, VELR finds the majority cluster that indicates the most dominant distribution of v^{y_i} as defined below:

$$k_{majority} = argmax_{k\in K}\, \pi_k$$

Let $v^{y_i}_{x,m}$ be an observation of v^{y_i}, which is $P^m(y_i|\boldsymbol{x})$. Then, like a normal clustering method, we assign each certainty $v^{y_i}_{x,m} = P^m(y_i|\boldsymbol{x})$ to a cluster k_i ($i \in \{1, 2\}$):

$$k(v^{y_i}_{x,m}) = argmax_{k\in K} \frac{\pi_k \mathcal{N}(v^{y_i}_{x,m} \,|\mu_k, \sigma_k)}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{N}(v^{y_i}_{x,m} \,|\mu_{k'}, \sigma_{k'})}$$

Our goal is to reject samples whose prediction is likely to be wrong. To make the model prediction more conservative, we hypothesize that the least confident certainty (i.e., posterior probability) should be taken. Therefore, for min-majority voting, the minimum $P^m(y_i|\boldsymbol{x})$ in the majority cluster is taken as the consensus prediction for the posterior probability, denoted as $P^*(y_i|\boldsymbol{x})$:

$$P^*(y_i|\boldsymbol{x}) = \min_{m\in MM^{y_i}_{x}} v^{y_i}_{x,m}$$

$$MM^{y_i}_{\boldsymbol{x}} = \{m : m \in M \text{ where } k(v^{y_i}_{x,m}) = k_{majority}\}$$

By taking the majority cluster, the value of $P^*(y_i|\boldsymbol{x})$ by min-majority voting is less likely to be zero.

2.2.2 Uniform voting

Uniform voting takes the mean of the certainty distribution per class y_i, $v^{y_i}_{x,m} = P^m(y_i|\boldsymbol{x})$:

$$P^*(y_i|x) = \frac{1}{M}\Sigma_{m\in M}\, v^{y_i}_{x,m}$$

Notice that uniform voting is equivalent to soft voting with the uniform weight of one (1.0) [9].

2.3 Rejecting Uncertain Predictions

Once the consensus certainty $P^*(y_i|\boldsymbol{x})$ is determined for each class y_i, a rejection method is applied. The rejection is made based on a hypothesis that a reliable prediction should agree with highly certain posterior probabilities across models.

Our rejection function $r(\boldsymbol{x})$ is defined with pre-defined threshold θ: $\mathbf{R}(0, 1)$ as:

$$r(\boldsymbol{x}) = \max_{y_i\in Y} P^*(y_i|\boldsymbol{x}) - \theta$$

The sample $\boldsymbol{x}$ is rejected if $r(\boldsymbol{x}) \leq 0$ and accepted otherwise. Therefore, our classification function $f(\boldsymbol{x})$ is:

$$f(x) = \begin{cases} Reject & if\ r(x) \leq 0 \\ argmax_{y_i\in Y} P^*(y_i|\boldsymbol{x}) & otherwise \end{cases}$$

Rejection increases the risk of not being able to make a prediction but decreases the risk of creating a wrong prediction. In some domains, including education, the quality of the model output is more important than the quantity, and often making a wrong prediction results in a harmful consequence. The task of pedagogical question generation, which is reposted later in section 4.2 as a sample task, is an example of such a sensitive task.

3. RELATED WORK

3.1 Training with Extremely Low Data Regime

Deep neural networks (DNN) are prone to overfit small training data. There has been extensive research conducted on preventing overfitting. Three commonly used techniques are: (1) restricting models and data, (2) pre-training models, and (3) augmenting data.

Restricting the model and data is used to prevent the model from being too complex. Regularization techniques are commonly used, including dropout [29], dropconnect [31], random noise [20, 22], and many others (for example, [11, 32]). Reducing the dimensionality of the input can also increase the generalizability of the model [1, 16]. However, it is not clear whether these regularization techniques work for extremely low data regimes.

Pre-training methods are used to initially train a model with data from a related task before fine-tuning the model using the target data. In NLP tasks, it is common to use pre-training models [8, 28, 35]. Although fine-tuning might be done with less amounts of data when a model is sufficiently pre-trained, it does not always work. Indeed, fine-tuning did not work for the question generation task that we used for an evaluation (section 4.2).

Data augmentation is conducted to increase the amount of training data. There are various methods proposed for DNN-based data augmentation [5, 14, 15, 18]. When unlabeled data are available, a generative technique model can be combined with semi-supervised learning [3, 12, 34]. These generative models might apply to extremely low data regimes. Zhang *et al.* [36] proposed a GAN-based data-augmentation technique, called DADA, specifically for extremely low data regimes. DADA involves a device called Augmenter that generates a new image given random noise and a label. DADA also involves a Discriminator, which acts as a classifier that outputs a binary decision for each class category, indicating whether the input belongs to the distribution of the real data for the target class.

Unlike the above-mentioned methods, VELR does not require changing a model structure or input data. Theoretically, VELR is thoroughly *data-agnostic*—it can be easily adapted to any classification or prediction tasks including NLP tasks. Practically, VELR should work as a reliable solution for many existing models with an extremely low data regime.

3.2 Classification with Rejection

For classification tasks that involve a high risk for misclassification, there has been research on classification with rejection, where a classifier may choose not to make a prediction in order to avoid wrong predictions [21]. The original study on classification with rejection [21] is based on a single model. It is not clear how classification with rejection works in conjunction with voting over an ensemble of overfitting models. The empirical study reported in the next section demonstrated that classification with rejection with voting shows a better performance than that with a single model in an extremely low data regime.

4. EVALUATION STUDY

An evaluation study was conducted to test the effectiveness of VELR. To validate the generality of the algorithm, VELR was applied to two different tasks—image classification and educational question generation. An NVIDIA GeForce RTX 3090 was used for the evaluation.

4.1 First task: Image classification

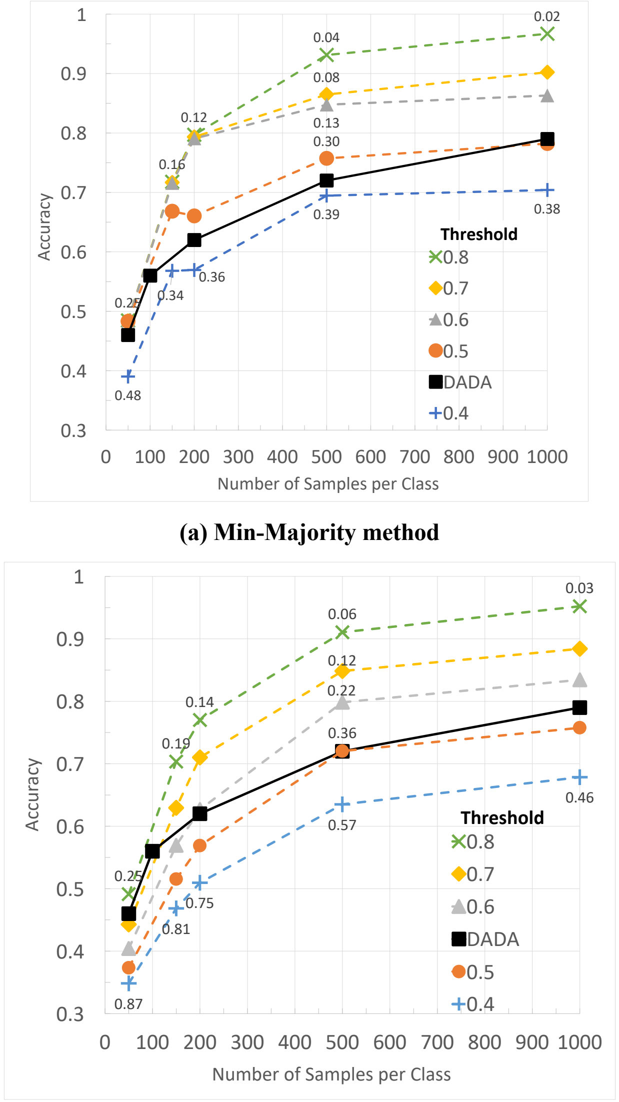
4.1.1 Method: Image classification

The first task used a subset of CIFAR-10 datasets [13] to simulate VELR being applied to an extremely low data regime.

CIFAR-10 contains 10 classes with 5000 samples per class. The training datasets we used consist of 50 (1% of complete training dataset), 150 (3%), 200 (4%), 500 (10%), and 1000 (20%) samples per class randomly sampled from the CIFAR-10 dataset.

To increase the reliability of the results, we created four different subsets of training data for each of the five different sample sizes mentioned above. The results reported below in the results section show the averaged performance among four subsets.

For each training subset, we trained 5000 models, applied VELR, and validated the ensemble outcome using the CIFAR-10 test dataset, which contains 10,000 samples.



(a) Min-Majority method

(b) Unform method

Figure 2. Comparison with DADA in terms of accuracy. Each line shows the change of accuracy (y-axis) with a given threshold θ depending on the number of training samples (x-axis). The value above each data point shows the predicted ratio (i.e., number of samples predicted without rejection / total number of samples).

The architecture of the classification model consists of two convolutional layers with max-pooling and three fully connected layers, as shown in Table 2 in Appendix. Each model was trained for 9000 steps. The batch size was 32. The learning rate was 10^{-3}. No regularization technique was used.

By applying VELR to this task, 10 consensus predictions $P^*(y_1|\boldsymbol{x})$, …, $P^*(y_{10}|\boldsymbol{x})$ were computed (cf. Figure 1).

The results were compared with a state-of-the-art model for ensemble learning with the extremely low data regimes, DADA [36]. Note that DADA uses data augmentation and regularization.

For this task, we also explored how the size of ensemble, i.e., the number of models trained, influences the performance of the classifier.

4.1.2 Results: Image classification

Figure 2 shows the accuracy of the prediction (y-axis) with different numbers of training data (x-axis). The accuracy was averaged over 4 trials. Since the standard deviation was smaller than 0.01 for all data points, it is not shown in the figure.

Figure 2-a shows results for min-majority voting, Figure 2-b shows uniform voting. Each line corresponds to a particular rejection threshold θ as shown in the legend. The numbers associated with a data point show the predicted ratio as defined as follows (not all data points show the predicted ratio for simplicity):

$$predicted\ ratio = \frac{\#\ samples\ predicted\ without\ rejection}{\#\ samples\ in\ the\ test\ data}$$

The figure only shows data with $0.4 < \theta < 0.8$, because there was a clear trend that the larger the θ, the higher the accuracy becomes regardless of other factors (e.g., size of data and voting method). Also, when the threshold became greater than 0.8, a considerable number of samples was rejected.

The figure shows that VELR with min-majority voting outperformed DADA when $\theta \geq 0.6$. VELR with uniform voting also outperformed DADA when $\theta \geq 0.7$. *The current data demonstrates that a very simple ensemble model with no data augmentation and regularization can outperform a complex model that includes a generative model for data augmentation.*

As shown in Figure 2 when the training data size was fixed (for example, see 500 per class), the larger the θ, the higher the accuracy but the lower the predicted ratio was. This indicates a trade-off between the accuracy and the predicted ratio. We therefore investigated the trade-off of each voting method as shown next.

We also plotted the trade-off between accuracy (y-axis) and the predicted ratio (x-axis), comparing training models with 200 (Figure 4-a in Appendix) and 1000 (Figure 4-b) samples per class. The plots clearly show a trade-off between accuracy and predicted ratio. Together with the fact that threshold and accuracy are negatively correlated, this finding suggests that *when the threshold is increased, the accuracy also increases at the cost of predicted ratio (or the number of rejections)*. Figure 4 also shows that uniform voting was clearly better than a single model prediction, and consistently better than or equal to min-majority voting. Because of this, we used uniform voting for the second task as shown in the next section.

4.2 Second task: Educational Question Generation

The task of generating educational questions motivated us to develop the VELR method. This section describes the overview of the question generation model that we developed and why we needed to invent VELR.

4.2.1 Model to be trained: Question generation

As part of our on-going effort to develop evidence-based learning-engineering methods that facilitate the creation of online courseware, called PASTEL [17], we developed a system for automated question generation, called QUADL [27]. A unique characteristic of QUADL is that it is aimed to generate a question for a key concept in a given didactic text that is assumed to help students attain a specific learning objective. The input to QUADL is a didactic text and a learning objective, and the output is a pair of a question and an answer.

QUADL consists of two machine-learning models: (1) An answer prediction model that identifies a key token in a given didactic text that is related to a specific learning objective. (2) A question conversion model that converts the didactic text that contains the key token into a question for which the key token is the literal answer. Notice that the answer for the generated question can be literally identified in the source didactic text. Since the source didactic text is sampled from the actual online courseware, the generated questions, by definition, are verbatim questions.

The technical details of the models used in QUADL is provided elsewhere [27]. Here, we provide a quick overview of those models sufficient to understand how the ensemble technique VELR was applied to train QUADL.

Given a pair of a learning objective ***LO*** and a sentence ***S***, QUADL generates a question ***Q*** that is assumed to be suitable to achieve the learning objective ***LO*** (Figure 5 in Appendix shows an overview of QUADL). The following is an example of ***LO***, ***S***, and ***Q***:

Learning objective (*LO*): Describe the basic (overall) structure of the human brain.
Sentence (*S*): The dominant portion of the human brain is the cerebrum.
Question (*Q*): What is the dominant portion of human brain?
Answer (*A*): cerebrum

Notice that the target token is underlined in the sentence ***S*** and becomes the answer ***A*** for the question ***Q***.

The input of the answer prediction model is a single sentence ***S*** (or a "source sentence" for the sake of clarity) and a learning objective ***LO***. The output from the answer prediction model is a target token index <*Is*, *Ie*>, where *Is* and *Ie* show the index of the start and end of the target token within the source sentence ***S*** relative to the learning objective ***LO***. The models may output <*Is*=0, *Ie*=0>, indicating that the source sentence is not suitable to generate a question for the learning objective.

For the *answer prediction model*, we adopted Bidirectional Encoder Representation from Transformers (BERT) [8]. The final hidden state of the BERT model is fed to two single layer classification models. One of them outputs a vector of probabilities $Ps(i)$ indicating the probability that the i-th token in the sentence is the beginning of the target token. Likewise, another classification model outputs a vector of probabilities that the end index is located at the j-th token, $Pe(j)$. To compute the probability of a target token index <$Is=i$, $Ie=j$>, a normalized sum of $Ps(i)$ and $Pe(j)$ is first calculated as the joint probability $P(Is=i, Ie=j)$ for every possible span ($Is < Ie$) in the sentence. The probability $P(Is=0, Ie=0)$ is also computed, which indicates the likelihood that the sentence is not suitable to generate a question for the learning objective. The index <$Is=i$, $Ie=j$> with the largest joint probability becomes the final prediction.

For the *question conversion model*, we hypothesize that if a target token is identified in a source sentence, a pedagogically valuable question can be generated by converting that source sentence into a verbatim question using a sequence-to-sequence model that can generate fluent and relevant questions. Therefore, we decided to use the state-of-the-art technology, called ProphetNet [23], for now. ProphetNet is an encoder-decoder pre-training model that is optimized by future n-gram prediction while predicting n-tokens simultaneously.

4.2.2 Methods: Question generation

Training QUADL models. For the current study, QUADL was applied to an existing online course "Anatomy and Physiology" (A&P) hosted on the Open Learning Initiative (OLI) at Carnegie Mellon University. The A&P course consists of 490 pages and has 317 learning objectives. To create training data for *the answer prediction model*, in-service instructors who actively teach the A&P course manually tagged the didactic text. The instructors were asked to tag each sentence ***S*** in the didactic text to indicate the target tokens relevant to specific learning objective ***LO***.

A total of 8 instructors generated 350 pairs of <***LO***, ***S***> for monetary compensation. Those 350 pairs of token index data were used to fine-tune the answer prediction model. As expected, fine-tuning the BERT model with only 350 training data points resulted in severe overfit—in average, only 38% of predicted target tokens were correct relative to the ground truth data (i.e., 350 pairs of <***LO***, ***S***>). *VELR was then applied to training the answer prediction model to overcome the model overfit.*

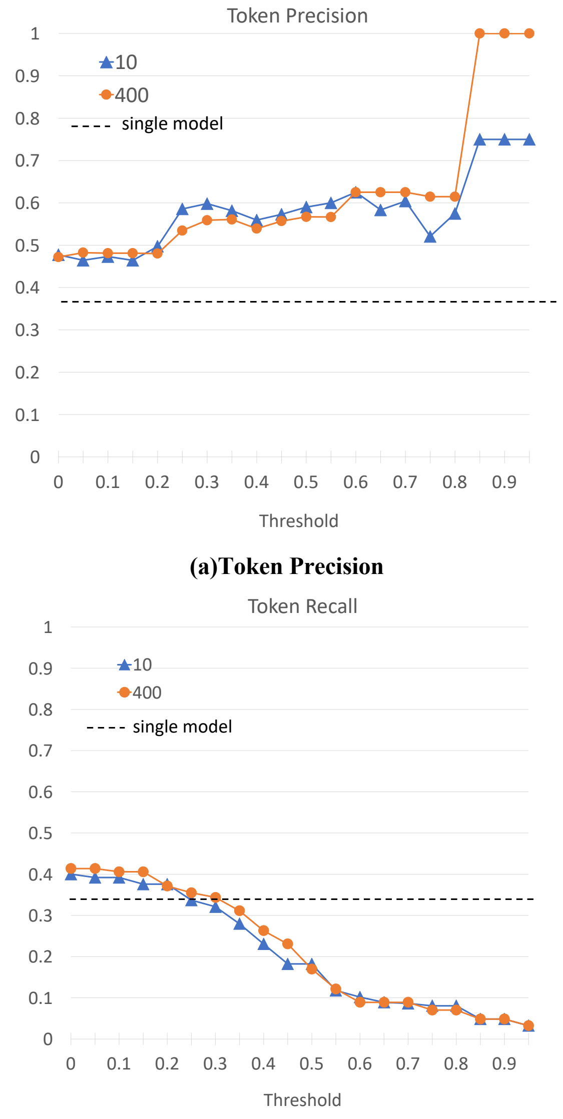
To make an ensemble prediction, 400 answer prediction models were trained independently using the same training data, but each with a different parameter initialization. Using all 400 answer prediction models, an ensemble model prediction was made as follows.

To begin with, recall that for each answer prediction model AP_k ($k = 1, ..., 400$), two vectors of probabilities are output, one for the start index $Ps^k(i)$, and another one for the end index $Pe^k(j)$. Uniform voting was then applied for each vector. That is, those probabilities were averaged across all models to obtain the ensemble predictions $Ps^*(i)$ and $Pe^*(j)$ for the start and end indices, respectively. The final target token prediction $P^*(Is=i, Ie=j)$ was then computed using $Ps^*(i)$ and $Pe^*(j)$ as described in section 4.2.1.

In the current study, we used threshold of 0.4 for rejection because otherwise the accuracy of the model is too low (token precision <0.60) or the recall is too small (token recall < 0.20) on the test dataset. How the token precision and the token recall were computed is described in section 4.2.3

For the question conversion model, we used an existing instance of ProphetNet that was already trained on the SQuAD1.1 dataset [24], one of the most commonly used datasets for question generation tasks that contains question-answer pairs retrieved from Wikipedia.

Generating questions using QUADL. Once trained, QUADL was applied to the pages of OLI A&P courseware (excluding pages that were used in the training dataset for the answer prediction model). A total of 2191 questions were generated from 490 pages with 317 learning objectives.



(a)Token Precision

(b) Token Recall

Figure 3. Average of token precision (a) and token recall (b) when VELR is used with 10 models (blue triangle markers) and 400 models (orange round markers). The dashed line (black) shows token precision and token recall by a single answer prediction model with no rejection.

State-of-the-art question generation model. We used Info-HCVAE [6], a state-of-the-art question generation model, as a baseline. Info-HCVAE generates questions without taking a learning objective into account. Instead, it extracts key concepts from a given paragraph and generates questions for them. Therefore, our primary motivation to use Info-HCVAE as a baseline (besides its outstanding performance at the time of writing this paper) is to compare question generation with and without taking learning objectives into account. The details of the evaluation of question generation are beyond the scope of this paper but can be found in [27].

Survey. Five in-service instructors who actively teach the OLI A&P course (the "participants" hereafter) were recruited for a survey study. The survey contained 100 items, each consisting of a paragraph, a learning objective, a question, and an answer.

Participants were asked to rate the *prospective pedagogical value* of proposed questions using four evaluation metrics on a 5-point Likert scale that we developed for the current study: answerability, correctness, appropriateness, and adoptability.

Answerability refers to whether the question can be answered from the information shown in the proposed paragraph. Correctness is whether the proposed answer adequately addresses the question. Appropriateness is whether the question is appropriate for helping students achieve the corresponding learning objective. Adoptability is how likely the participants would adapt the proposed question to their class.

Each individual participant rated all 100 survey items. The questions used in the survey were created either by QUADL, Info-HCVAE, or a human expert. There were 34 questions generated by QUADL, 33 questions by Info-HCVAE, and 33 human-generated questions from the same OLI A&P course. Since the survey did not mention the source of the included questions, the participants blindly evaluated the prospective pedagogical value of those questions.

Consequently, five responses per question were collected, which is notably richer than any other human-rated study for question generation in the current literature, as these studies often involve only two coders.

4.2.3 Results: Question generation

Our primary research questions regarding the use of VELR with QUADL are: (1) How does VELR improve the accuracy (token precision) of the answer prediction? (2) How pedagogically adequate are the questions generated by QUADL when combined with VELR?

Accuracy of Answer Prediction Model. To investigate how VELR improved the accuracy of the answer prediction model used in QUADL, we evaluated the token precision with different threshold values.

We operationalized the accuracy of target token identification using two metrics: token precision and token recall. *Token precision* is the number of correctly predicted tokens divided by the number of tokens in the prediction. *Token recall* is the number of correctly predicted tokens divided by the number of ground truth tokens. For example, suppose a sentence "*The target tissues of the nervous system are muscles and glands*" has the ground truth tokens as "*muscles and glands*." When the predicted token is "glands," the token precision is 1.0 and recall is 0.33**.**

Figure 3 shows the change of token precision (a) and token recall (b) depending on the threshold when VELR is applied on 10 answer prediction models vs. 400 models. The figure shows the aggregated average over 7 runs.

Figure 3-a shows that VELR improves the token precision of the answer prediction model. When VELR is not used, the average token precision was 0.38 (as shown in the black dashed line). When VELR was used with a threshold of 0.6, for example, the token precision was 0.63.

There was a trade-off between precision and recall as predicted. As Figure 3-b shows the token recall decreased when the threshold increased. The plots in the figure also suggest that there was no significant difference between 10 models and 400 models when unified voting was applied.

In sum, *VELR improved the performance of the answer prediction model (which is based on the BERT architecture) even when it was trained with only 350 data points*. For uniform voting, the number of models did not significantly impact the performance of the ensemble model. Due to the rejection, there is a clear trade-off between the soundness (token precision) and the completeness (token recall) of the ensemble model prediction.

As discussed before, the use of VELR is beneficial for tasks where soundness is valued over completeness—for pedagogical question generation, it is far more useful to generate a small number of pedagogically valuable questions than to generate lots of harmful questions. So, a further research question is: How pedagogically adequate are the questions generated by QUADL when combined with VELR?

Quality of the generated questions. The results on the answer prediction model shown above promisingly suggest that VELR has a practical application for generating questions for existing online courseware. The current survey results supported this expectation. Table 1 shows the survey results.

To see if there was a difference in ratings between questions with the different sources (QUADL vs. Infor-HCVAE vs. Human), a one-way ANOVA was applied separately to each metric. The results revealed that source is a main effect for ratings on all four metrics; F(2, 97) = 36.38, 24.15, 26.11, and 25.03, for answerability, correctness, appropriateness, and adoptability, respectively. A post hoc analysis using Tukey's test showed that there was a statistically significant difference between QUADL and Info-HCVAE; *t*(97)=1.87, 1.50, 1.52, 1.39 for each metric, p < 0.05 for all metrics. There was, however, no significant difference between QUADL and human-generated questions for each of the four metrics: *t*(97)=0.40, 0.25, 0.16, 0.25, p = 0.19, 0.53, 0.78, 0.45 respectively.

In sum, the results from the current survey data suggest that *QUADL-generated questions were evaluated as on-par with human-generated questions when VELR is applied to the answer prediction model trained with an extremely small data regime*.

We further investigated how the consensus certainty of ensemble prediction of the answer prediction model impacted the quality of the generated questions. We sampled a subset of questions used in the survey by excluding the questions whose source target sentences would have been rejected if a threshold higher than 0.4 had been applied. In other words, we investigated the following research question: How does the rejection threshold used by VELR when applied to the answer prediction model impact the ratings of the QUADL-generated questions? We plotted how the ratings change if thresholds higher than 0.4 were applied (**Figure 6** in Appendix). The figure shows a trend that the participants would have increased their rating when higher values for rejection threshold were used, though the differences were relatively small and not monotonic.

Table 1. Survey results. Average rating by five participants (± standard deviations). The rating values range from 1 as strongly disagree to 5 as strongly agree. The rejection threshold for the answer prediction model was set to 0.4.

	QUADL	Human	Info-HCVAE
Answerability	4.19 ± 0.74	3.79 ± 0.89	2.32 ±1.15
Correctness	4.05 ± 0.72	3.80 ± 0.83	2.55±1.21
Appropriateness	4.04 ± 0.74	3.88 ± 0.76	2.52±1.25
Adoptability	3.79 ± 0.62	3.53 ± 0.78	2.39±1.10

5. DISCUSSION AND LIMITATIONS

Building a valid prediction model with extremely low data regimes is an omnipresent challenge in education research and many other domains when human annotation is required. Therefore, developing a data-agnostic technique to overcome this issue is vital to advance the pragmatic theory of learning engineering.

We proposed a voting function based on the distribution of the predicted posterior probability (or "certainly"). The experiment with CIFAR-10 showed that both min-majority and uniform voting can achieve better accuracy than the state-of-the-art method, DADA [36], even without any regulation or data augmentation technique on the image classification task.

Although concepts of soft-voting and classification with rejection have already been studied in the current literature, VELR is the first in the literature that combines soft-voting technique with rejection to carry out *ensemble learning to overcome the issue of overfitting when a model is trained with an extremely low data regime*.

In this paper, we explored only the Gaussian mixture model for min-majority voting, there are various ways to implement a voting technique by fitting different probability distributions. We conjecture that using a voting technique that better estimates a distribution of the posterior probability will further expand the potential of the proposed ensemble method.

We demonstrated that VELR is useful for a real-world application: pedagogical question generation as a learning-engineering tool for online courseware creation. However, the observations related to the evaluation of VELR on QUADL needs some attention. Since the total number of QUADL-generated questions used in the survey is small (34) due to the cost of the human-evaluation, the number of questions included in a subset when a higher threshold was applied was significantly small, too (Figure 6 in Appendix). The survey study should be replicated with a larger number of questions to further validate the current findings.

6. CONCULSION

We found that combining soft voting among overfitting models and rejection based on the distribution of the learned posterior probability leads to remarkable accuracy on tasks even when models were trained with extremely low data regimes and were hence severely overfit.

While a conventional solution for overfitting due to extremely low data regimes is to restrict the flexibility of the model or increase the amount of data using the data-augmentation techniques, proposed VELR (Voting-based Ensemble Learning with Rejection) applies to any task and any models that estimate predicted certainly using posterior probability. VELR combines multiple overfitting models to output reliable predictions rather than preventing a model from overfitting while training.

The extremely low data regime is one of the most common problems in many practical tasks including educational data mining. Yet, building a reliable machine-learning model with a limited amount of data is an unavoidable demand. Further research to study the theoretical foundation for overcoming the overfitting problem under an extremely low data regime is therefore needed.

7. ACKNOWLEDGEMENTS

The research reported here was supported by National Science Foundation Grant No. 2016966 and No.1623702 to North Carolina State University.

8. REFERENCES

[1] Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics, 2*(4), 433-459.

[2] Amin-Naji, M., Aghagolzadeh, A., & Ezoji, M. (2020). CNNs hard voting for multi-focus image fusion. *Journal ofAmbient Intelligence and Humanized Computing, 11*(4), 1749-1769.

[3] Anaby-Tavor, A., Carmeli, B., Goldbraich, E., Kantor, A., Kour, G., Shlomov, S., . . . Zwerdling, N. (2020). *Do not have enough data? Deep learning to the rescue!* Paper presented at the Proceedings of the AAAI Conference on Artificial Intelligence.

[4] Chrysos, G. G., Kossaifi, J., & Zafeiriou, S. (2020). Rocgan: Robust conditional gan. *International Journal of Computer Vision, 128*(10), 2665-2683.

[5] Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. V. (2019). *Autoaugment: Learning augmentation strategies from data.* Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[6] Dai, Z., Yang, Z., Yang, F., Cohen, W. W., & Salakhutdinov, R. R. (2017). Good semi-supervised learning that requires a bad gan. *Advances in neural information processing systems, 30*.

[7] Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological), 39*(1), 1-22.

[8] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[9] Domingos, P. (2000). *Bayesian averaging of classifiers and the overfitting problem.* Paper presented at the ICML.

[10] Dong, J., & Lin, T. (2019). Margingan: Adversarial training in semi-supervised learning. *Advances in neural information processing systems, 32*.

[11] Ghosh, R., & Motani, M. (2021). Network-to-Network Regularization: Enforcing Occam's Razor to Improve Generalization. *Advances in neural information processing systems, 34*.

[12] Kingma, D. P., Mohamed, S., Jimenez Rezende, D., & Welling, M. (2014). Semi-supervised learning with deep generative models. *Advances in neural information processing systems, 27*.

[13] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.

[14] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems, 25*.

[15] Li, C., Xu, T., Zhu, J., & Zhang, B. (2017). Triple generative adversarial nets. *Advances in neural information processing systems, 30*.

[16] Liu, B., Wei, Y., Zhang, Y., & Yang, Q. (2017). *Deep Neural Networks for High Dimension, Low Sample Size Data.* Paper presented at the IJCAI.

[17] Matsuda, N., Shimmei, M., Chaudhuri, P., Makam, D., Shrivastava, R., Wood, J., & Taneja, P. (in press). PASTEL: Evidence-based learning engineering methods to facilitate creation of adaptive online courseware. In F. Ouyang, P. Jiao, B. M. McLaren, & A. H. Alavi (Eds.), *Artificial Intelligence in STEM Education: The Paradigmatic Shifts in Research, Education, and Technology* (pp. 1-16). New York, NY: CSC Press.

[18] Mikołajczyk, A., & Grochowski, M. (2018). *Data augmentation for improving deep learning in image classification problem.* Paper presented at the 2018 international interdisciplinary PhD workshop (IIPhDW).

[19] Miyato, T., & Koyama, M. (2018). cGANs with projection discriminator. *arXiv preprint arXiv:1802.05637*.

[20] Neelakantan, A., Vilnis, L., Le, Q. V., Sutskever, I., Kaiser, L., Kurach, K., & Martens, J. (2015). Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*.

[21] Ni, C., Charoenphakdee, N., Honda, J., & Sugiyama, M. (2019). On the calibration of multiclass classification with rejection. *Advances in neural information processing systems, 32*.

[22] Noh, H., You, T., Mun, J., & Han, B. (2017). Regularizing deep neural networks by noise: Its interpretation and optimization. *Advances in neural information processing systems, 30*.

[23] Qi, W., Yan, Y., Gong, Y., Liu, D., Duan, N., Chen, J., . . . Zhou, M. (2020, November). *ProphetNet: Predicting Future N-gram for Sequence-to-SequencePre-training.* Paper presented at the Findings of the Association for Computational Linguistics: EMNLP 2020, Online.

[24] Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

[25] Salman, S., & Liu, X. (2019). Overfitting mechanism and avoidance in deep neural networks. *arXiv preprint arXiv:1901.06566*.

[26] Sanyal, A., Dokania, P. K., Kanade, V., & Torr, P. H. (2020). How benign is benign overfitting? *arXiv preprint arXiv:2007.04028*.

[27] Shimmei, M., Bier, N., & Matsuda, N. (to appear). *Machine-Generated Questions Attract Instructors when Acquainted with Learning Objectives* Paper presented at the Proceedings of the International Conference on Artificial Intelligence in Education.

[28] Song, K., Tan, X., Qin, T., Lu, J., & Liu, T.-Y. (2020). Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems, 33*, 16857-16867.

[29] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research, 15*(1), 1929-1958.

[30] Van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning, 109*(2), 373-440.

[31] Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., & Fergus, R. (2013). *Regularization of neural networks using*

dropconnect. Paper presented at the International conference on machine learning.

[32] Yang, T., Zhu, S., & Chen, C. (2020). Gradaug: A new regularization method for deep neural networks. *Advances in neural information processing systems, 33*, 14207-14218.

[33] Yang, X., Song, Z., King, I., & Xu, Z. (2021). A survey on deep semi-supervised learning. *arXiv preprint arXiv:2103.00550.*

[34] Yang, Y., Malaviya, C., Fernandez, J., Swayamdipta, S., Bras, R. L., Wang, J.-P., . . . Downey, D. (2020). Generative data augmentation for commonsense reasoning. *arXiv preprint arXiv:2004.11546.*

9. APPENDIX

Classification model for CIFAR-10

Each model was trained for 9000 steps. The batch size was 32. The learning rate was 10^{-3}. No regularization technique was used.

Table 2. The architecture of a model used for the image classification task.

Layer [Output shape]
5*5 Conv. 2*2 Max-pooling [32, 6, 14, 14]
5*5 Conv. 2*2 Max-pooling [32, 16, 5, 5]
Fully connected ReLu [32, 120]
Fully connected ReLu [32, 84]
Fully connected [32, 10]
10-class Softmax [32, 10]

Trade-off between Accuracy and Predicted Ratio

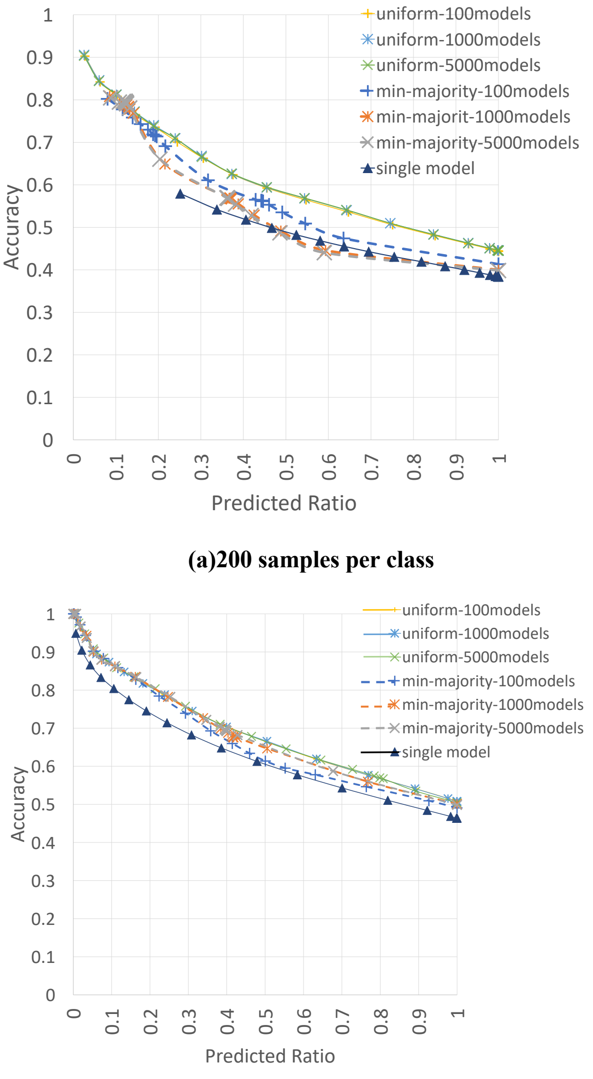
The dotted line and solid line show min-majority and uniform voting, respectively. Each voting schema has three plots with 100, 1000, and 5000 models as shown with different markers. Each line contains 20 data points (denoted as markers on the line). Each data point corresponds to a particular threshold ranging from 0.95 to 0.0 (i.e., no rejection), decreasing by 0.5. Since the predicted ratio increases as the threshold is lowered, the 20 data points on the line are coincidentally arranged in a decremental manner, from left to right, for the threshold (hence the threshold values are not displayed on the plot for simplicity). For example, the second marker from the right on min-majority models shows that when $\theta = 0.90$, the min-majority voting over 1000 models yielded the accuracy of 0.49 with the predicted ratio of 0.62.

The figure shows uniform voting was clearly better than a single model prediction, and consistently better than or equal to min-majority voting.

[35] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems, 32.*

[36] Zhang, X., Wang, Z., Liu, D., & Ling, Q. (2019). *Dada: Deep adversarial data augmentation for extremely low data regime classification.* Paper presented at the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

[37] Zhou, Z.-H. (2012). *Ensemble methods: foundations and algorithms*: CRC press.



(a)200 samples per class

(b)1000 samples per class

Figure 4. Trade-off between Accuracy and Predicted Ratio.

Overview of QUADL

The answer prediction model identifies start/end index <*Is, Ie*> of the target token (i.e., key term) in S. When S is not suitable for LO, it outputs <0,0>. The question conversion model converts S with target token to a verbatim question.

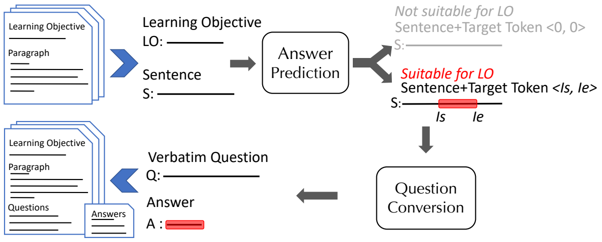


Figure 5. An overview of QUADL used for question generation task.

Change of Average Rating for Questions Generated by QUADL

Figure 6 was plotted to answer the research question: How does the rejection threshold used by VELR when applied to the answer prediction model impact the ratings of the QUADL-generated questions?

Each data point includes a subset of questions used in the survey excluding the questions whose source target sentences would have been rejected if a threshold higher than 0.4 had been applied.

The figure shows how the ratings change if thresholds higher than 0.4 were applied. The figure shows a trend that the participants would have increased their rating when higher values for rejection threshold were used, though the differences were relatively small and not monotonic. Appropriateness, for example, increased from 4.04 to 4.30 when the threshold was changed from 0.4 to 0.75. Accordingly, acceptability also increased from 3.53 to 4.10.

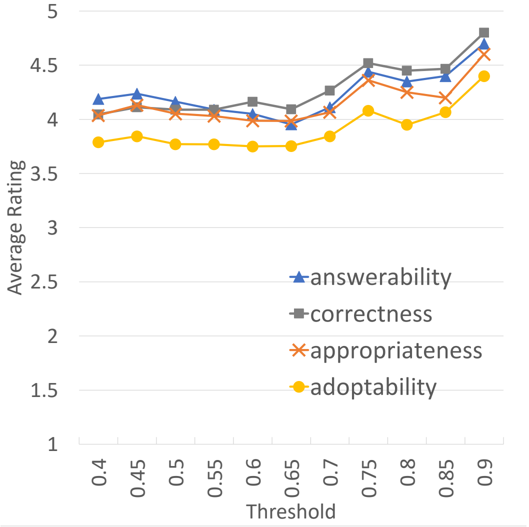


Figure 6. Change of average ratings with higher threshold VELR.

Knowledge Tracing Over Time: A Longitudinal Analysis

Morgan P Lee
Unity Hall, 27 Boynton Street
Worcester, MA, U.S.A.
mplee@wpi.edu

Ethan Croteau
Unity Hall, 27 Boynton Street
Worcester, MA, U.S.A.
ecroteau@wpi.edu

Ashish Gurung
Unity Hall, 27 Boynton Street
Worcester, MA, U.S.A.
agurung@wpi.edu

Anthony F. Botelho
1221 SW 5th Ave
Gainesville, FL, U.S.A.
abotelho@coe.ufl.edu

Neil T. Heffernan
Unity Hall, 27 Boynton Street
Worcester, MA, U.S.A.
nth@wpi.edu

ABSTRACT

The use of Bayesian Knowledge Tracing (BKT) models in predicting student learning and mastery, especially in mathematics, is a well-established and proven approach in learning analytics. In this work, we report on our analysis examining the generalizability of BKT models across academic years attributed to "detector rot." We compare the generalizability of Knowledge Training (KT) models by comparing model performance in predicting student knowledge within the academic year and across academic years. Models were trained on data from two popular open-source curricula available through Open Educational Resources. We observed that the models generally were highly performant in predicting student learning within an academic year, whereas certain academic years were more generalizable than other academic years. We posit that the Knowledge Tracing models are relatively stable in terms of performance across academic years yet can still be susceptible to systemic changes and underlying learner behavior. As indicated by the evidence in this paper, we posit that learning platforms leveraging KT models need to be mindful of systemic changes or drastic changes in certain user demographics.

Keywords

Bayesian Knowledge Tracing, Longitudinal Analysis, Student Modeling, Generalizability, Detector Rot

1. INTRODUCTION

Modeling student knowledge and mastery of particular skills is a foundational problem to the domain of learning analytics and its intersections with education and artificial intelligence. The first proposed solution to the Knowledge Tracing (KT) problem, dubbed Bayesian Knowledge Tracing (BKT) by its creators [3], modeled knowledge as the mastery of multiple independent knowledge concepts (KCs, or skills) and estimated mastery through the use of a latent variable in a Hidden Markov Model. Student mastery of a skill is assumed to be a noisy representation of this latent variable, moderated by four parameters: a student's prior knowledge, the likelihood of mastering the skill through attempting a problem, the chance a student answers correctly by guessing, and the chance a student answers incorrectly by mistake. Future work augmenting BKT attempted to improve model performance by modifying the assumptions of the initial model. For example, classical BKT models assume the acquisition of knowledge is unidirectional, from a state of non-mastery to a state of mastery. Relaxing this assumption and allowing for student knowledge to move bidirectionally between mastery and non-mastery resulted in models that more accurately predict student performance, and thus more accurately model student knowledge [14]. Further model extensions include allowing individual students to have personal prior knowledge rates [10] and giving individual questions their own guess and slip rates [11]. While other statistical models such as Performance Factors Analysis [12] showed initial promise, later advances in the domain of machine learning resulted in the creation of deep learning models to solve the problem of KT, utilizing a recurrent neural network in Deep Knowledge Tracing (DKT) [13] and self-attention in Self Attentive Knowledge Tracing (SAKT) [9]. However, BKT still serves as a useful way of modeling student knowledge due to the model's interpretability, especially in comparison to larger models [6]. BKT models require far fewer parameters to train in comparison to the deep-learning models even when BKT models incorporate the available extensions. If the performance of the model is a priority and the generalizability of the model is not guaranteed, then training new models in response to some population shift is advisable. Indeed, this is a common practice in online learning platforms when such shifts occur, such as the beginning of a new school year or the integration of a new curriculum. However, how do we know how often our KT models should be retrained?

More precisely, we wish to examine the performance of BKT models across time. Our analysis was guided by the following research questions:

RQ1. Do BKT models lose predictive power with time?

RQ2. Does the complexity of a KT model impact its generalizability through time?

RQ3. Do sudden shifts in student populations or behavior

M. Lee, E. Croteau, A. Gurung, A. F. Botelho, and N. T. Heffernan. Knowledge tracing over time: A longitudinal analysis. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 296–301, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115788

impact model performance?

To answer these questions, we gathered data collected through the ASSISTments platform across four school years from 2018-2022. We then compare model performance on data from the same year as training with model performance across years. Additionally, we posit that the COVID-19 pandemic caused a shift in student and teacher perception of technology for learning as there were no alternatives available to adopting technology in classrooms. As such we examine the shift in the learner behavior by examining the generalizability KT models trained on pre-pandemic data to predict learning during the pandemic and vice versa. We begin by discussing the challenges to education posed by the COVID-19 pandemic, focusing on the rapid adoption of online learning tools during the pandemic. Next, we describe the data generation and sampling process for our analysis. The student data available from ASSISTments across the four academic years establish a fair comparison of the KT models that is not susceptible to the size of the dataset since different academic years had varying number of users. We then describe the KT models used in our analysis and the approach we took in examining the generalizability of KT models. We compare model performance of classical BKT and BKT with forgetting models within the same academic year, across different academic years, and across the beginning of the pandemic, along with the impact of the forgetting parameter on model generalizability. We then discuss the implications of our findings on the implementation of KT models, and discuss the limitations of our analysis and their implications for future research.

1.1 COVID-19 Pandemic

The COVID-19 pandemic has presented many challenges to the delivery of education to students [4]. As many schools closed their doors, students were required to attend classes and complete coursework using online tools. This resulted in the rapid adoption of online learning platforms leading to a significant growth in the user base of platforms such as ASSISTments. This influx of new users likely introduces a more diverse group of students into school populations, since schools integrated various learning tools to support their students. Additionally, the sudden shift in the perception of technology and its use in teaching for many schools also present an interesting opportunity to explore the robustness and generalizability of KT models.

Given the wide-reaching changes to education caused by the COVID-19 pandemic, the impact these changes had on student learning requires more investigation. For the purposes of our analysis, we divided data gathered into two meta-groups: pre-pandemic and post-pandemic, with ”post-pandemic” data merely denoting data that was gathered after the initial transition into online learning in mid-March 2020.

2. RELATED WORK

Analysis of more complex inferential models used by MATHia found that models intended to detect “gaming the system” behaviors [2] trained on older data were significantly less precise on newer data [7]. It was found that more contemporary machine learning models designed to detect gaming experienced a greater performance decrease than classical, computationally simpler models. This phenomenon was called ”detector rot” by its authors in reference to a similar phenomenon called ”code rot” in which code performance decreases over time [5]. The analysis provided by [7] featured a comparison of models trained on data collected more than a decade apart, with models trained to solve a complex problem with a large feature space. We aim to contribute to the understanding of detector rot by examining model performance along more granular time steps, across dramatic population shifts, and with models solving a problem with a much smaller feature space.

3. METHODS

3.1 Data Collection

Data for each school year was gathered from problem logs between the dates of September 1st and June 1st. Summer months were excluded as the student population during the summer can vary more drastically from year to year. The student cohort during some summers primarily consists of students requiring additional work to reach their credit requirements while other summers are filled with high achieving students working on extra credit. Problem level data from the typical academic year was then filtered based on several criteria in order to ensure different academic years were able to be directly compared. Comparison between two populations with little intersection in the skills being assessed would result in poor model generalizability based solely on underfitting. To ensure direct comparisons were possible and appropriate, we limited our underlying populations to problems sourced from the two most popular open-source math curricula available through OER [8] on the ASSISTments platform: EngageNY/Eureka Math and Kendall Hunt’s Illustrative Mathematics. From these two curricula, we calculated the top five hundred most commonly assigned problem sets across all four of our target years. The final populations we constructed before sampling were filtered by these top five hundred common problem sets, with the exception of the 2018-2019 school year. Data from this year was significantly more sparse than other years due to the introduction of a new implementation of the ASSISTments tutor, and as such we only applied the curriculum filter to this year. Since the introduction of the new tutor experience, student behavior has been logged in a consistent fashion.

3.2 Student Modeling

Students in ASSISTments can make unlimited attempt when answering a problem until they answer it correctly, with the number of attempts a student takes to correctly answer a problem being recorded in problem-level data. The problem level data also includes information on the number of help requests and if the student requested for the answer to the problem. BKT attempts to predict student performance on attempts to apply a skill [3]. However, in the original problem level data, each student/problem interaction only has a single row. In an effort to encode information about how many attempts a student took to complete a problem, the original problem logs were used to create a dataset with each row representing a student’s attempt to apply a skill. Additionally, if a student’s final correct answer for a question came from a bottomed-out hint, explanation,

Table 1: Dataset Information

Year	Total Rows	Total Assignments	Unique Students	% Correct
2018-2019	291,437	31,930	4,425	0.534
2019-2020	521,781	130,173	47,595	0.526
2020-2021	8,459,566	1,310,652	190,366	0.494
2021-2022	2,645,324	361,546	58,216	0.547

Table 2: Feature List

Feature	Description
user	Unique student identifier
assignment	Unique identifier for an assignment
correct	0 if the student incorrectly applies skill, 1 otherwise
start_time	Timestamp of when the problem was started by the student
problem	Unique identifier for a problem
curriculum	Curriculum the problem originated from
skill	Skill being assessed by the current question
attempt_number	Counts which attempt on the problem this row represents

or simply requesting the answer, the student's final correct answer was treated as an incorrect application of the skill. Information about the amount of data available for each year at the end of the filtering and encoding process can be found in Table 1, while a description of the available features present in all datasets can be found in Table 2. Ten samples of 25,000 assignment level data per year were generated for each year of the data. To investigate the effect of additional model parameters on model generalizability, two models were trained at each step: one with forgetting and one without. Other than this additional parameter, all training parameters were initialized in the same way. Models were constructed using pyBKT, a Python library for creating BKT models described by [1]. For analysis of within-year performance, a five-fold cross-validation was performed on each sample from the 10 samples, resulting in fifty measurements of AUC being taken for exploring model performance within the training year. For the inter-year performance analysis, the models were trained on one of the 10 random samples from a target year and evaluated on the other corresponding random samples from the other three years. This resulted in the generation of thirty measurements of AUC, since the model for each year was trained on 10 random samples and tested on 10 random samples from other three years resulting in 30 data points for the across year generalizability analysis. Finally, data from the 18-19, 19-20, and 20-21 years was split around the beginning of the COVID-19 pandemic (the precise date was March 12, 2020) and ten samples each containing 50,000 assignment level data were generated on each side of this split. The same process of five-fold cross-validation followed by a cross-year train/test analysis was performed on these pandemic samples.

4. RESULTS

4.1 Robustness Over Time (RQ1)

Data gathered from our evaluations across academic years can be found in Tables 3 and 4, while the resulting means from our five-fold cross-validations plotted along with their 95% confidence intervals can be found in Figure 4.2. Rather unsurprisingly, the within year generalizability of the BKT models was high with the BKT + forgetting model always outperforming the classical BKT model. However the model generalizability when trained on one year and applied to other years varied across academic years: by comparing the training year averages provided in Tables 3 and 4, models trained on the 20-21 and 21-22 school years had higher average AUCs, while the 18-19 school year produced the least generalizable models. Similarly, different years were easier to generalize to than others, with the 18-19 school year having a much lower testing year average for both model types.

4.2 Complexity (RQ2)

One general observation seen from each of the analyses is that BKT+Forgets consistently outperforms classical BKT in terms of its predictive power as measured by mean AUC. Our findings strongly suggest the introduction of a forgetting parameter for each skill can be done with little chance of significantly harming a model's later generalizability.

4.3 Sudden Shifts: Pandemic Analysis (RQ3)

Data gathered from training and evaluating models before and after the COVID-19 pandemic can be found in Table 5, while these means and relevant confidence intervals were plotted in Figure 4.3. Models trained on data gathered before the pandemic had difficulties generalizing to post-pandemic data. Consider models evaluated on the post-pandemic dataset. The delta means between models trained on pre-pandemic data and post-pandemic data were 0.022 for classical BKT and 0.028 for BKT + forgets. This generalization problem also occurs when considering models evaluated on the pre-pandemic data, suggesting that KT models are susceptible to losses in predictive power following major shifts in underlying user populations.

As was true with the year-by-year data, the addition of a forgetting parameter to the classical BKT model significantly improves performance, even across the population shift. The use of model additions may improve generalizability in a way that can withstand significant shifts in population and user behavior.

Table 3: BKT cross-year analysis

	18-19 Data	19-20 Data	20-21 Data	21-22 Data	Training Year Avg
18-19 Model		0.669	0.672	0.678	0.673
19-20 Model	0.682		0.729	0.714	0.709
20-21 Model	0.686	0.726		0.734	0.715
21-22 Model	0.690	0.724	0.748		0.721
Testing Year Avg	0.686	0.706	0.716	0.709	

Table 4: BKT+Forgets cross-year analysis

	18-19 Data	19-20 Data	20-21 Data	21-22 Data	Training Year Avg
18-19 Model		0.687	0.683	0.694	0.688
19-20 Model	0.686		0.740	0.730	0.719
20-21 Model	0.706	0.739		0.757	0.734
21-22 Model	0.708	0.736	0.766		0.735
Testing Year Avg	0.700	0.721	0.730	0.727	

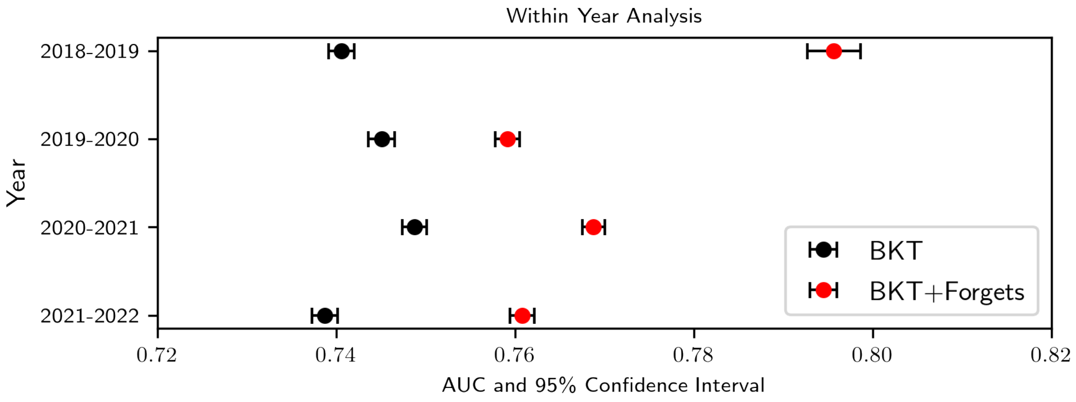
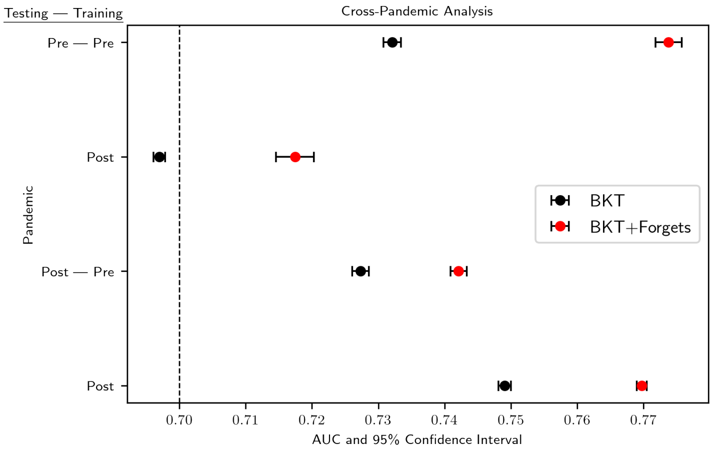


Table 5: Cross-Pandemic Analysis

Testing Period	Training Period	Model Type	Mean AUC	95%CE
Pre-pandemic	Pre-Pandemic	BKT	0.732	[0.731,0.733]
		BKT+Forgets	0.774	[0.772,0.776]
	Post-Pandemic	BKT	0.697	[0.696,0.698]
		BKT+Forgets	0.717	[0.715,0.720]
Post-pandemic	Pre-pandemic	BKT	0.727	[0.726,0.729]
		BKT+Forgets	0.742	[0.741,0.743]
	Post-pandemic	BKT	0.749	[0.748,0.750]
		BKT+Forgets	0.770	[0.769,0.771]



5. DISCUSSION

In this paper, we explored the generalizability of KT models within and across academic years. The concept of ”detector rot” [7] is a recent addition to how we understand inferential models and their applications in online tutoring platforms. With this analysis of how KT models perform over time, we intend to further explore the concept as it applies to KT models. Our exploration began by collecting data in a way that ensured the set of skills in each year's worth of data were comparable and then translating the raw problem level data into attempt-level representations of student performance. Models were evaluated both on the year in which they were trained (by a five-fold cross-validation), and on the other available years. We trained both classical BKT models and models with a forgetting parameter to investigate how adding model parameters impacts model generalizability. We also divided our available data around the beginning of the COVID-19 pandemic to investigate the impact of sudden shifts in population size on model generalizability. We have a few key findings to report from these investigations. (a) In contrast to more sophisticated models, BKT's performance is relatively stable from year to year, indicating that the problem of detector rot is far less prevalent within the domain of KT. (b) The addition of forgetting parameters to BKT models consistently improves performance across multiple years of student population drift, and across more sudden changes of population. (c) Drastic changes in an online tutoring system's user base can impact BKT models' performance.

While our results indicate KT model stability over short-term population changes, our work is limited by several factors which future research could address. Our attempts to ensure each dataset contained a large overlap of skills could result in our models showing higher AUCs across time than comparable KT models would show in a product-scale system. Also, the 18-19 school year was particularly difficult for other models to generalize to. This is likely due to the sparsity of data for that year limiting our ability to filter by commonly assigned problem sets. Future work leveraging more data as ASSISTments continues to be used through time may give more insight as to why some years are easier for models to generalize to than others. Our analysis of RQ2 was also limited by only exploring how forgetting parameters impact generalizability. Future work incorporating more extensions to BKT, such as those described by [10] and [11], or utilizing more complex KT models like PFA [12] and DKT [13] is required to investigate trade-offs between model complexity and generalizability found in previous detector rot research [7]. Finally, while our analysis of RQ3 shows that BKT models had trouble generalizing across the beginning of the COVID-19 pandemic, the reasons for this could be numerous, including the sparsity of data pre-pandemic compared to post-pandemic or differences in student behavior after the pandemic began. Further analysis of how the COVID-19 pandemic impacted student behavior, possibly focusing on the transitional period from remote schooling back to in-person learning, could provide more insight into how student demographic changes affect KT models.

Acknowledgements

Dr. Heffernan thanks NSF (e.g., 2118725, 2118904, 1950683, 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, & 1535428), IES (e.g., R305N210049, R305D210031, R305A170137, R305A170243, R305A180401, R305A120125 & R305R220012), GAANN (e.g., P200A180088 & P200A150306), EIR (U411B190024 & S411B210024), ONR (N00014-18-1-2768), and Schmidt Futures. None of the opinions expressed here are that of the funders.

6. REFERENCES

[1] A. Badrinath, F. Wang, and Z. Pardos. pybkt: an accessible python library of bayesian knowledge tracing models. *arXiv preprint arXiv:2105.00385*, 2021.

[2] R. S. Baker, A. T. Corbett, K. R. Koedinger, and A. Z. Wagner. Off-task behavior in the cognitive tutor classroom: When students” game the system”. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 383–390, 2004.

[3] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[4] W. E. Forum. The rise of online learning during the covid-19 pandemic | world economic forum, 2020.

[5] C. Izurieta and J. M. Bieman. A multiple case study of design pattern decay, grime, and rot in evolving software systems. *Software Quality Journal*,

21(2):289–323, 2013.

[6] M. Khajah, R. V. Lindsey, and M. C. Mozer. How deep is knowledge tracing? *CoRR*, abs/1604.02416, 2016.

[7] N. Levin, R. S. Baker, N. Nasiar, S. Fancsali, and S. Hutt. Evaluating gaming detector model robustness over time. In *Proceedings of the 15th International Conference on Educational Data Mining, International Educational Data Mining Society*, 2022.

[8] C. o. C. S. S. O. National Governors Association Center for Best Practices. Common core state standards (mathematics standards), 2010.

[9] S. Pandey and G. Karypis. A self-attentive model for knowledge tracing. *arXiv preprint arXiv:1907.06837*, 2019.

[10] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *International conference on user modeling, adaptation, and personalization*, pages 255–266. Springer, 2010.

[11] Z. A. Pardos and N. T. Heffernan. Kt-idem: Introducing item difficulty to the knowledge tracing model. In *International conference on user modeling, adaptation, and personalization*, pages 243–254. Springer, 2011.

[12] P. I. Pavlik Jr, H. Cen, and K. R. Koedinger. Performance factors analysis–a new alternative to knowledge tracing. *Online Submission*, 2009.

[13] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[14] Y. Qiu, Y. Qi, H. Lu, Z. A. Pardos, and N. T. Heffernan. Does time matter? modeling the effect of time with bayesian knowledge tracing. In *EDM*, pages 139–148, 2011.

Towards Generalizable Detection of Urgency of Discussion Forum Posts

Valdemar Švábenský
University of Pennsylvania
svabenskyv@gmail.com

Ryan S. Baker
University of Pennsylvania
ryanshaunbaker@gmail.com

Andrés Zambrano
University of Pennsylvania
afzambrano97@gmail.com

Yishan Zou
University of Pennsylvania
amieezou@gmail.com

Stefan Slater
University of Pennsylvania
slater.research@gmail.com

ABSTRACT

Students who take an online course, such as a MOOC, use the course's discussion forum to ask questions or reach out to instructors when encountering an issue. However, reading and responding to students' questions is difficult to scale because of the time needed to consider each message. As a result, critical issues may be left unresolved, and students may lose the motivation to continue in the course. To help address this problem, we build predictive models that automatically determine the urgency of each forum post, so that these posts can be brought to instructors' attention. This paper goes beyond previous work by predicting not just a binary decision cut-off but a post's level of urgency on a 7-point scale. First, we train and cross-validate several models on an original data set of 3,503 posts from MOOCs at University of Pennsylvania. Second, to determine the generalizability of our models, we test their performance on a separate, previously published data set of 29,604 posts from MOOCs at Stanford University. While the previous work on post urgency used only one data set, we evaluated the prediction across different data sets and courses. The best-performing model was a support vector regressor trained on the Universal Sentence Encoder embeddings of the posts, achieving an RMSE of 1.1 on the training set and 1.4 on the test set. Understanding the urgency of forum posts enables instructors to focus their time more effectively and, as a result, better support student learning.

Keywords

educational data mining, learning analytics, text mining, natural language processing, forum post urgency

1. INTRODUCTION

In computer-supported learning environments, students often ask questions via email, chat, forum, or other communication media. Responding to these questions is critical for learners' success since students who do not receive a timely reply may struggle to achieve their learning goals. In a small-scale qualitative study of online learning [11], students who received delayed responses to their questions from the instructor reported lower satisfaction with the course. Another study showed that students who received instructor support through personalized emails performed better on both immediate quizzes and delayed assessments [15].

Massive Open Online Courses (MOOCs) are a prevalent form of computer-supported learning. MOOCs enable many students worldwide to learn at a low cost and in a self-paced environment. However, many factors cause students to drop out of MOOCs, including psychological, social, and personal reasons, as well as time, hidden costs, and course characteristics [22].

A MOOC's discussion forum is central to decreasing the risk of student drop-out since it promotes learner engagement with the course. Students use the forum to ask questions, initiate discussions, report problems or errors in the learning materials, interact with peers, or otherwise communicate with the instructor. Andres et al. [5] reviewed studies on MOOC completion and discovered that certain behaviors, such as spending above-average time in the forum or posting more often than average, are associated with a higher likelihood of completing the MOOC. Similarly, Crues et al. [10] showed that students who read or write forum posts are more likely to persist in the MOOC. At the same time, instructor participation in the forum and interaction with students promotes engagement with the course [25].

For the reasons above, the timely response of instructors to students' posts is important. In a study with 89 students, 73 of them preferred if the instructor responded to discussion forum posts within one or two days [16]. However, this is not always feasible. Students' posts that require an instructor's response may be unintentionally overlooked due to MOOCs' scale. Instructors can feel overwhelmed by a large number of posts and often lack time to respond quickly enough or even at all. As a result, issues that students describe in the forum are left unsolved [4], leaving the learners discouraged and frustrated.

1.1 Problem Statement

Since MOOCs tend to have far more students than other computer-supported learning environments, identifying urgent student questions is crucial. We define urgency in

V. Švábenský, R. Baker, A. Zambrano, Y. Zou, and S. Slater. Towards generalizable detection of urgency of discussion forum posts. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 302–309, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115790

discussion forum posts as the degree of how quickly the instructor's response to the post is needed. Urgency is expressed on an ordinal scale from 1 (not urgent at all) to 7 (extremely urgent). This scale is adopted from the Stanford MOOCPosts data set [2], arguably the most widely used publicly available data set of MOOC discussion forum posts. It contains 29,604 anonymized, pre-coded posts that have been employed in numerous past studies (see Section 2).

Educational data mining and natural language processing techniques may allow us to automatically categorize forum posts based on their urgency. Our goal is to build models that will perform such categorizations to determine whether a timely response to a post would be valuable. Ultimately, we aim to help instructors decide how to allocate their time where it is needed the most.

Automatically determining the urgency of forum posts is a challenging research problem. Since posts highly vary in content – the students can type almost anything – the data may contain a lot of noise that is not indicative of urgency. In addition, it is difficult to generalize the trained models to other contexts because of linguistic differences caused by different variants of English or by non-native speakers of English, as well as terms that are highly specific to a course topic.

1.2 Contributions of This Research

We collected and labeled an original data set of 3,503 forum posts, which we used to train and cross-validate several classification and regression models. From the technical perspective, we tested two different families of features and compared the performance of the regressors, multi-class classifiers, and binary classifiers.

Subsequently, we tested the generalizability of the results by using the independent Stanford MOOCPosts data set [2] of 29,604 forum posts as our holdout test set.

2. RELATED WORK

Almatrafi et al. [3] used the Stanford MOOCPosts data set to extract three families of features: Linguistic Inquiry and Word Count (LIWC) attributes, term frequency, and post metadata. They represented the problem of urgency prediction as binary classification, considering the post not urgent if it had a label below 4, and urgent for 4 and above. The study evaluated five classification approaches: Naive Bayes, Logistic Regression, Random Forest, AdaBoost, and Support Vector Machines. The best-performing model was AdaBoost, able to classify the forum post urgency with the weighted F1-score of 0.88.

Sha et al. [20] systematically surveyed approaches for classifying MOOC forum posts. They discovered that previous research used two types of features: textual and metadata. Textual features consist of n-grams, post length, term frequency-inverse document frequency (TF-IDF), and others. Metadata features include the number of views of the post, the number of votes, and creation time. Furthermore, the survey compared six algorithms used to construct urgency models from these features, building on the methods by Almatrafi et al. [3]. Four traditional machine learning (ML) algorithms included Naive Bayes, Logistic Regression, Random Forest, and Support Vector Machines. The best results were yielded by combining textual and metadata features and training a Random Forest model (AUC = 0.89, F1 = 0.89). Two deep learning algorithms examined in the survey were CNN-LSTM and Bi-LSTM. Using the same metrics, these models performed even better than the traditional ones. However, in their follow-up work, Sha et al. [21] concluded that deep learning does not necessarily outperform traditional ML approaches overall. The best urgency classifier, again a Random Forest model, achieved an F1-score of 0.90 (AUC was not reported).

Several studies employed the Stanford MOOCPosts data set to train a neural network (NN) for identifying urgent posts. Capuano and Caballé [7] created a 2-layer feed-forward NN on the Bag of Words representation of the posts, reaching an F1-score of 0.80. Alrajhi et al. [4] used a deep learning model that combined text data with metadata about posts. They reported an F1-score of 0.95 for predicting non-urgent posts (defined by labels 1–4) and 0.74 for predicting urgent posts (label > 4). Yu et al. [24] also transformed the problem into binary classification. They compared three models, the best being a recurrent NN achieving an F1-score of 0.93 on non-urgent posts and 0.70 on urgent posts.

More advanced approaches include those by Guo et al. [12], who proposed an attention-based character-word hybrid NN with semantic and structural information. They achieved much higher F1-scores overall, ranging from 0.88 to 0.92. Khodeir [14] represented the Stanford MOOCPosts data set using BERT embeddings and trained gated recurrent NNs to predict the posts' urgency. The best model achieved weighted F1-scores from 0.90 to 0.92.

Previous work used the Stanford MOOCPosts data set to train the models but did not evaluate them on other data. Therefore, the models may overfit to that data set but be ineffective in other contexts. By training models on our own data and testing it on the Stanford MOOCPosts data set, we provide a new perspective within the current body of work in post urgency prediction. We aim to achieve a more generalizable modeling of forum posts' urgency and provide valuable information for instructors who support large numbers of learners.

In doing so, we also build upon work by Wise et al. [23], who researched techniques for determining which MOOC forum posts are related content-wise. They used the Bag of Words representation of posts and extracted unigrams and bigrams as features. Using a Logistic Regression model, they reached an accuracy between 0.73 and 0.85, depending on the course topic. We use similar methods but for a different purpose.

In designing responses to urgent posts, it is valuable to consider the work by Ntourmas et al. [17], who analyzed how teaching assistants respond to students' forum posts in two MOOCs. The researchers combined content, linguistic, and social network analysis to discover that teaching assistants mostly provide direct answers. The researchers suggested that this approach does not adequately promote problem-solving. Instead, they argued that more indirect and guiding approaches could be helpful.

3. RESEARCH METHODS

This section describes the data and approaches used to train and evaluate predictive models of forum post urgency.

3.1 Data Collection and Properties

We collected posts from students who participated in nine different MOOCs at the University of Pennsylvania (UPenn) from the years 2012 to 2015. The nine MOOCs focused on a broad range of domains (in alphabetical order): accounting, calculus, design, gamification, global trends, modern poetry, mythology, probability, and vaccines. This breadth of covered topics enables us to prevent bias towards certain course topics and support generalization across courses.

To construct the research data set, we started by randomly sampling 500 forum posts for each of the nine courses. Then, we removed posts that:

- were in a language other than English
- contained only special symbols and characters
- contained only math formulas
- contained only website links

As a result, we ended up with 3,503 forum posts from 2,882 students. This data set included a similar number of posts from each course (between 379 and 399 per course), adding up to the total of 3,503.

Each data point consists of three fields: a unique numerical student ID, the timestamp of the forum post submission, and the post text. All remaining post texts are in the English language, though not all students who wrote them were native speakers of English. The posts contain typos, grammatical errors, and so on, which we did not correct.

3.2 Data Anonymization

To preserve student privacy, two human readers manually redacted personally identifiable information in the posts. The removed pieces of text included names of people or places, contact details, and any other information that could be used to determine who a specific poster was.

Each of the two readers processed roughly half of the post texts from each of the nine courses (195 posts per course per reader on average). The split was selected randomly.

After this anonymization procedure was completed, the data were provided to the research team. To support the replicability of our results, the full data set used in this research can be found at https://github.com/pcla-code/forum-posts-urgency.

Since we use only de-identified, retrospective data, and the numerical student IDs cannot be traced back to the students' identity, this research study received a waiver from the university's institutional review board.

3.3 Data Labeling

Three human coders (distinct from those individuals who anonymized the data) manually and independently labeled the 3,503 anonymized post texts. To ensure the approach was unified, they completed coder training and followed a predefined protocol that specified how to assign an urgency label to each post. The protocol is available alongside our research data at https://github.com/pcla-code/forum-posts-urgency.

The three coders initially practiced on a completely separate data set of 500 labeled posts with the urgency label hidden. After each coded response, they revealed the correct label and consulted an explanation if they were off by more than 1 point on the scale.

At the end of the training, we computed the inter-rater reliability of each coder within the practice set. Specifically, we calculated continuous (i.e., weighted) Cohen's Kappa using linear weighting. The three coders achieved the Kappa of 0.57, 0.49, and 0.56, respectively. We note that the weighted values are typically lower than regular Kappa. For instance, weighted Kappa values are lower when there is a relatively large number of categories [6], as is seen in our data sets. They are also lower in cases where, for example, one coder is generally stricter than another (i.e., different means by coder) even though their ordering of cases is identical [19].

When the coders felt confident in coding accurately, the study coordinator sent them 20 different posts from the separate data set with the urgency label removed. If they coded them accurately, they received a batch of 50 original posts (out of our 3,503 collected) for actual coding. In case a coder was unsure, discrepancies were resolved by discussion.

As stated in Section 1.1, we use the term *urgency* to indicate how fast an instructor should respond to the post. For example, if a post is very urgent, then the instructor or teaching assistant (TA) should respond to it as soon as possible. If a post is not urgent, then the instructor and TA might not have to respond to the post at all. Degrees of urgency were mapped to ordinal scores proposed by Agrawal and Paepcke [2] (and later adopted by related work [3, 4]) as follows:

- 1: No reason to read the post
- 2: Not actionable, read if time
- 3: Not actionable, may be interesting
- 4: Neutral, respond if spare time
- 5: Somewhat urgent, good idea to reply, a teaching assistant might suffice
- 6: Very urgent: good idea for the instructor to reply
- 7: Extremely urgent: instructor definitely needs to reply

Example for label 1: *"Hi my name is [REDACTED] and I work in the healthcare industry, looking forward to this course!"*

Example for label 5: *"When will the next quiz be released? I'd like to get a head start on it since I've got some extra time these days."*

Table 1: Distribution of training labels in each course. The row *Train* is the sum of all the label frequencies in the individual courses. The row *Test* is the distribution of the labels in the separate test set (rounded up, see Section 3.6).

Course	1	2	3	4	5	6	7
Accoun	199	63	18	53	48	6	0
Calcul	64	167	44	88	31	2	0
Design	148	114	36	31	35	15	1
Gamif	243	62	15	31	28	0	0
Global	123	197	21	16	25	5	0
Modern	131	214	30	15	7	1	0
Mythol	129	149	59	24	24	5	0
Probab	125	115	48	72	31	5	0
Vaccin	114	139	63	43	21	9	1
Train	1276	1220	334	373	250	48	2
Test	3501	14997	3308	3054	2259	2471	14

Example for label 7: *"The website is down at the moment, [link] seems down and I'm not able to submit the Midterm. Still have the "Final Submit" button on the page, but it doesn't work. Are the servers congested?"*

Table 1 lists the frequencies of individual urgency labels in the training data across each of the nine courses, as well as their total count. We also detail the frequencies of urgency labels in our test set (see Section 3.6). As the table shows, the frequencies of the labels differ between the training and test set; thus, if our models perform well in this case, they are likely to be robust when predicting data with various distributions.

3.4 Data Automated Pre-Processing

Before training the models, we performed automated data cleaning and pre-processing that consisted of the following steps in this order:

- Converting all text in the posts to lowercase.
- Replacing all characters, except the letters of the English alphabet and numbers, with spaces.
- Removing duplicate whitespace.
- Removing common stopwords in the English language, such as articles and prepositions.
- Stemming, that is, automatically reducing different grammatical forms of each word to its root form [13].

Each pre-processed post contained 51 words on average (stdev 76, min 1, max 1390).

3.5 Model Training and Cross-Validation

The problem of assigning a forum post into one of seven ordered categories corresponds to multi-class ordinal classification or regression (Section 3.5.1). In addition, we also converted the problem to binary classification (Section 3.5.2) to provide a closer comparison with related work.

3.5.1 *Multi-class Classification and Regression*

We hypothesized that regression algorithms would be more suitable for our use case because they can capture the order on the 1–7 scale, which categorical classifiers cannot achieve. We used a total of six classification and regression algorithms:

- Random Forest (RF) classifier,
- eXtreme Gradient Boosting (XGB),
- Linear Regression (LR),
- Ordinal Ridge Regression (ORR),
- Support Vector Regression (SVR) with a Radial Basis Function (RBF) kernel, and
- Neural Network (NN) regressor.

We used Python 3.10 and standard implementations of the algorithms in the Scikit-learn module [18], using TensorFlow [1] and Keras [9] for the neural networks. The Python code we wrote to train and evaluate the models is available at https://github.com/pcla-code/forum-posts-urgency.

All algorithms had default hyperparameter values provided by Scikit-learn. The only exception was the neural network with the following settings discovered experimentally:

- Input layer with 128 nodes, 0.85 dropout layer, and ReLU activation function,
- One hidden layer with 128 nodes, 0.85 dropout layer, and ReLU activation function,
- Output layer with 1 node and ReLU activation function.

Each algorithm was evaluated on two families of features: one based on *word counts* (Bag of Words or TF-IDF representations of the forum post texts), the other based on *Universal Sentence Encoder v4* (USE) [8] numerical feature embeddings of the forum post texts.

During model training, we used 10-fold student-level cross-validation in each case. The metrics chosen to measure classification/regression performance were Root Mean Squared Error (RMSE) and Spearman ρ correlation between the predicted and actual values of urgency on the validation set. We chose Spearman instead of Pearson correlation because the urgency labels are ordinal data. The output of the regression algorithms was left as a decimal number, i.e., we did not round it to the nearest whole number.

3.5.2 *Binary Classification*

In addition, we trained separate models for binary classification. Following the precedent from the related work [4], the urgency label was converted to 0 if it was originally between 1–4, and converted to 1 if it was originally larger than 4. We did not adopt the approach of Almatrafi et al. [3], who considered a post urgent if it was labeled 4 or above, since based on the scale description defined by Agrawal and Paepcke [2] (see Section 3.3), we do not consider "Neutral" posts to be urgent. (When we tried doing this, it caused only a slight improvement in the model performance.)

Then, we trained RF, XGB, and NN classifier models. The performance evaluation metrics were macro-averaged AUC ROC and weighted F1-score.

3.6 Model Generalizability Evaluation

To determine the generalizability of our models, we evaluated them on held-out folds of the training set, then tested them on the Stanford MOOCPosts data set. This data set is completely separate from the training and validation sets and should, therefore, indicate how well our models would perform in different courses and settings.

The test set uses the 1–7 labels but with .5 steps, meaning that some posts can be labeled as 1.5 or 6.5, for example. We did not round these during model training to verify generalizability across both types of labels. However, when labeling our training set, we did not consider .5 labels since the coders felt it added too much granularity. Earlier work did not explicitly differentiate the .5 labels from the integers.

4. RESULTS AND DISCUSSION

This section details the results from both families of models: one based on word counts and the other on Universal Sentence Encoder. Then, we compare our models with those from related literature.

4.1 Models with Word Count Features

These models used the Bag of Words or TF-IDF representations of the forum post texts.

4.1.1 Multi-class Classification and Regression

We tested the following combinations of settings and hyperparameters for the word count models on the training and cross-validation set:

- Method of feature extraction. TF-IDF performed slightly better than Bag of Words.
- Range of n-grams extracted from the data. We tried unigrams, bigrams, and a combination of the two. The best results were obtained when using unigrams only. Models based on bigrams only or those that combined unigrams and bigrams performed worse. In the 3,503 posts, we had 774 unigram and 226 bigram features.
- Minimal/maximal allowed document frequency for each term. Here, the best-performing cut-off was to discard the bottom/top 1% of extreme document frequencies, so the ranges were set to 0.01 and 0.99, respectively. Using this approach made the algorithms run substantially faster, but given the extreme cut-offs, it did not appreciably change the values. Without setting the cut-offs, the training of some models took several hours.
- Feature unitization. It either did not impact or slightly worsened the model performance in all cases, so we did not use it.

Table 2 summarizes the performance of all models. Support vector regression performed best overall on the training and cross-validation set in terms of both metrics: RMSE and Spearman ρ correlation. It also outperformed the other approaches on the separate test set.

Table 2: Performance of multi-class classification/regression models on the training set of 3,503 posts (UPenn) and the test set of 29,604 posts (Stanford). Features: word counts.

	Training and cross-validation set		Different university test set	
Model	RMSE	ρ	RMSE	ρ
RF	1.3550	0.4258	1.7781	0.2676
XGB	1.3338	0.4326	1.7419	0.3086
LR	1.2385	0.4419	(large)	0.3432
ORR	1.1501	0.4750	1.4229	0.3484
NN	1.1269	0.4897	1.4395	0.3746
SVR	1.0946	0.5503	1.4138	0.3982

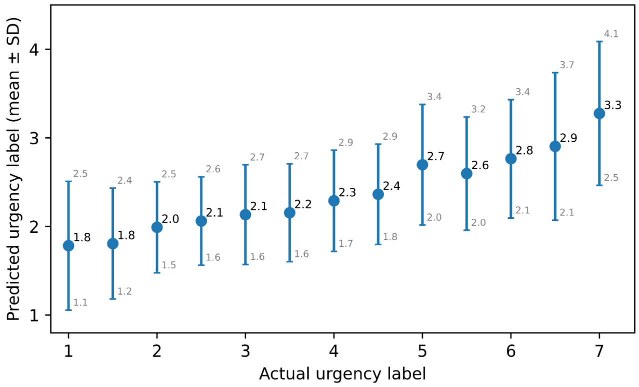


Figure 1: Prediction results of the best performing model (SVR) on the separate test set using the word count features.

Figure 1 shows the predictions of the best model on the test set. Most urgency labels are under-predicted, but they are still predicted in the increasing order of urgency, which demonstrates that the model is detecting the ranking.

After SVR, other regressors followed, with neural networks being the second best. Overall, the classifier models performed more poorly than the regression models. We expected this result since the urgency classes are ordinal, and the categorical classifiers cannot capture their ordering.

4.1.2 Binary Classification

Table 3 summarizes the performance of all models. The NN outperformed the remaining two classifiers, though the differences in AUC are more visible than for F1-score compared to XGBoost. Although the fit of RF and NN is nondeterministic, the results did not change substantially when we re-ran the model training multiple times.

When considering the prediction of non-urgent posts only, all models achieved a very high F1-score between 0.9512 (NN) and 0.9589 (RF) on the training set, and 0.8924 (NN) to 0.8971 (XGBoost) on the test set.

For the urgent posts only, the predictive power was much lower: between 0.1841 (RF) and 0.4168 (NN) on the training set, and 0.0025 (RF) to 0.2761 (NN) on the test set.

Due to the imbalance in favor of the non-urgent class, exper-

Table 3: Performance of binary classification models on the training set of 3,503 posts (UPenn) and the test set of 29,604 posts (Stanford). Features: word counts.

	Training and cross-validation set		Different university test set	
Model	AUC	F1	AUC	F1
RF	0.5522	0.8926	0.5005	0.7263
XGB	0.6178	0.9053	0.5412	0.7590
NN	0.6687	0.9055	0.5735	0.7759

imenting with decision cut-offs lower than the default 50% visibly improved the RF and XGBoost models' AUC (up to 0.7771) but improved the F1-score only slightly. The best results were achieved for decision thresholds of 10 or 15%.

4.2 Models with Feature Embeddings Using the Universal Sentence Encoder (USE)

4.2.1 Multi-class Classification and Regression

Table 4 summarizes the performance of all models. Again, SVR performed best on the training set, followed by NN. After that, other regressors and classifiers followed in the same order as with the word-count-based models. However, for the test set, while SVR still obtained the best ρ, it had slightly worse RMSE than the other three regressors. Overall, the model quality was better for USE than for TF-IDF.

Figure 2 shows the predictions made by the best model on the test set, with the trend being similar to Figure 1.

Table 4: Performance of multi-class classification/regression models on the training set of 3,503 posts (UPenn) and the test set of 29,604 posts (Stanford). Features: USE embeddings.

	Training and cross-validation set		Different university test set	
Model	RMSE	ρ	RMSE	ρ
RF	1.4707	0.3452	1.8995	0.2723
XGB	1.3569	0.4418	1.7753	0.3145
LR	1.1758	0.4717	1.3953	0.3882
ORR	1.1448	0.4983	1.3723	0.3964
NN	1.1045	0.5361	1.3988	0.4202
SVR	1.0956	0.5716	1.4065	0.4283

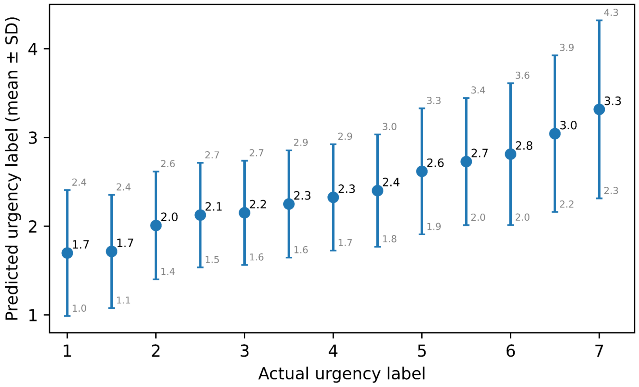


Figure 2: Prediction results of the best performing model (SVR) on the separate test set using the USE features.

Table 5: Performance of binary classification models on the training set of 3,503 posts (UPenn) and the test set of 29,604 posts (Stanford). Features: USE embeddings.

	Training and cross-validation set		Different university test set	
Model	AUC	F1	AUC	F1
RF	0.5094	0.8774	0.5002	0.7260
XGB	0.5863	0.9020	0.5246	0.7470
NN	0.6409	0.9054	0.5684	0.7760

4.2.2 Binary Classification

Table 5 summarizes the performance of all models. Compared to using the TF-IDF features, the results are surprisingly slightly worse, even though the differences are minimal in some cases. The overall order of models is preserved – again, the NN outperformed the other two models.

As previously, we observed similar imbalances in F1-scores when predicting non-urgent and urgent posts separately. For non-urgent posts, all models achieved a high F1-score between 0.9544 (NN) and 0.9597 (XGBoost) on the training set, and 0.8954 (RF) to 0.8974 (XGBoost) on the test set.

For predicting the urgent posts only, the predictive power is much lower: between 0.0366 (RF) and 0.3799 (NN) on the training set, and 0.0007 (RF) to 0.2563 (NN) on the test set. Again, the respective performance of the individual classifiers corresponds to the case with word count features.

As expected, decreasing the decision cut-off below 50% again substantially improved the overall model performance. The best results were again achieved for decision thresholds of 10 or 15%.

4.3 Comparison with the Results Published in Previous Literature

We now compare our results with the binary classification models reported in Section 2, which were trained on the Stanford MOOCPosts data set. We cannot compare our multi-class classification and regression analyses to past work since it treated this problem only as binary classification.

Almatrafi et al. [3] and Sha et al. [20] slightly differed from our approach in using the label 4 as the cut-off for post urgency, as opposed to 4.5. The best model by Almatrafi et al. [3], an AdaBoost classifier, achieved a weighted F1-score of 0.88. Our binary classifiers slightly outperformed this model, even though we used fewer types of features. This indicates that combining features from various sources does not necessarily improve model quality. Sha et al. [20] reported a RF model that scored F1 = 0.89 and AUC = 0.89. While we achieved similar F1-scores, our AUC was much lower. This could have been caused by the smaller training set, in which the class imbalance had a larger effect.

The NN approaches by Capuano and Caballé [7], Guo et al. [12], and Khodeir [14] reported F1-scores ranging from 0.80 to 0.92. Even though our NN models were much simpler and trained on a smaller data set, they achieved a similarly high F1 of 0.91.

Finally, Alrajhi et al. [4] and Yu et al. [24] reported the model performance separately for non-urgent and urgent posts. When considering non-urgent posts only, they reached F1-scores of 0.95 and 0.93, respectively. Our best-performing model on this task achieved F1 = 0.96 on the training set (RF, word count features) and 0.90 on the test set (XGBoost, USE features). When considering urgent posts only, they reported F1-scores of 0.74 and 0.70. Here, our models scored much worse, 0.42 on the training set and 0.28 on the test set (both approaches used NN on the word count features). The AUC scores were not reported in this case.

Overall, we achieved comparable or even slightly better performance in most cases. In addition, we evaluated the models for multi-class classification and regression, which the previous work did not consider.

We could not fully replicate past work because the feature set and the code used to produce the previous results were unavailable. This prevented us from testing the prior work on our data set, which would have helped to establish the generalizability of those earlier approaches.

4.4 Opportunities for Future Work

In future work, the urgency rating of forum posts can also be treated as a ranking problem. Using an ML algorithm, posts can be sorted from the most to the least urgent instead of classifying them as high or low priority. Even among the posts with the same urgency level, some messages should be addressed first. Therefore, reframing the problem to ranking learning would lead to a different model that suggests the most urgent post to address instead of estimating the level of urgency. Our current approach shows that regardless of the regression outputs, regressor models such as the SVR correctly estimate a higher urgency for more urgent posts. For this reason, ML models could show promising results for sorting the posts based on their urgency.

In addition, the post labeling scale could be improved, perhaps by simplifying it to fewer categories. In this study, we adopted the scale from previous work [2], used additionally in [3, 4] in order to be able to study the generalizability of findings across data sets. Finally, experimenting with over- or undersampling of the training set using algorithms such as SMOTE might improve model performance for certain labels.

To ensure even a higher degree of generalizability, future research could validate the models on data from different populations than those employed in our paper.

5. CONCLUSION

Responding to students' concerns or misunderstandings is vital to support students' learning in both traditional and MOOC courses. Since instructors cannot read all forum posts in large courses, selecting the posts that urgently require intervention helps focus instructors' attention where needed.

The presented research aims to automatically determine the urgency of forum posts. We used two separate data sets with different distributions and different approaches to the urgency scale (using .5 values or not) to support generalizability. Support vector regression models showed the highest performance in almost all aspects and cases. The best model from both categories of features (word count or numerical embeddings) performed similarly, with Universal Sentence Encoder embeddings being slightly better.

The results of this work can contribute to supporting learners and improving their learning outcomes by providing feedback to instructors and staff managing courses with large enrollment. The model quality has implications for practical use. Based on the RMSE values, it is unlikely that a highly urgent post will be labeled non-urgent and vice versa. From a practical perspective, implementing the urgency rating into MOOC platforms or large courses would help instructors, for example, by providing automated notification on posts with high urgency. In this case, however, students should not be aware of the inner workings of such a system. This is to prevent abuse by writing words with certain phrases to trigger instructor notifications.

6. ACKNOWLEDGMENTS

This research was supported by the National Science Foundation (NSF) (NSF-OAC#1931419). Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF.

We also thank Nathan Levin and Xiaodan Yu for their work in preparing and labeling the research data set.

7. REFERENCES

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: a system for large-scale machine learning. In *12th USENIX symposium on Operating Systems Design and Implementation (OSDI)*, pages 265–283, 2016. https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf.

[2] A. Agrawal and A. Paepcke. The Stanford MOOC Posts Dataset, 2014. http://infolab.stanford.edu/~paepcke/stanfordMOOCForumPostsSet.tar.gz. Original page available on URL: https://web.archive.org/web/20220908024430/https://datastage.stanford.edu/StanfordMoocPosts/.

[3] O. Almatrafi, A. Johri, and H. Rangwala. Needle in a haystack: Identifying learner posts that require urgent response in MOOC discussion forums. *Computers & Education*, 118:1–9, 2018. https://doi.org/10.1016/j.compedu.2017.11.002.

[4] L. Alrajhi, K. Alharbi, and A. I. Cristea. A Multidimensional Deep Learner Model of Urgent Instructor Intervention Need in MOOC Forum Posts. In V. Kumar and C. Troussas, editors, *Intelligent Tutoring Systems*, pages 226–236, Cham, 2020. Springer International Publishing. https://doi.org/10.1007/978-3-030-49663-0_27.

[5] J. M. L. Andres, R. S. Baker, D. Gašević, G. Siemens, S. A. Crossley, and S. Joksimović. Studying MOOC Completion at Scale Using the MOOC Replication Framework. In *Proceedings of the 8th International*

Conference on Learning Analytics and Knowledge, LAK '18, page 71–78, New York, NY, USA, 2018. Association for Computing Machinery. https://doi.org/10.1145/3170358.3170369.

[6] H. Brenner and U. Kliebsch. Dependence of weighted kappa coefficients on the number of categories. *Epidemiology*, 7:199–202, 1996. https://www.jstor.org/stable/pdf/3703036.pdf.

[7] N. Capuano and S. Caballé. Multi-attribute Categorization of MOOC Forum Posts and Applications to Conversational Agents. In L. Barolli, P. Hellinckx, and J. Natwichai, editors, *Advances on P2P, Parallel, Grid, Cloud and Internet Computing*, pages 505–514, Cham, 2020. Springer International Publishing. https://doi.org/10.1007/978-3-030-33509-0_47.

[8] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, April 2018. https://arxiv.org/abs/1803.11175.

[9] F. Chollet et al. Keras: The Python deep learning library, 2015. https://keras.io/.

[10] R. W. Crues, G. M. Henricks, M. Perry, S. Bhat, C. J. Anderson, N. Shaik, and L. Angrave. How Do Gender, Learning Goals, and Forum Participation Predict Persistence in a Computer Science MOOC? *ACM Trans. Comput. Educ.*, 18(4):1–14, 2018. https://doi.org/10.1145/3152892.

[11] A. Després-Bedward, T. Avery, and K. Phirangee. Student perspectives on the role of the instructor in face-to-face and online learning. *International Journal of Information and Education Technology*, 8(10):706–712, 2018. https://doi.org/10.18178/ijiet.2018.8.10.1126.

[12] S. X. Guo, X. Sun, S. X. Wang, Y. Gao, and J. Feng. Attention-Based Character-Word Hybrid Neural Networks With Semantic and Structural Information for Identifying of Urgent Posts in MOOC Discussion Forums. *IEEE Access*, 7:120522–120532, 2019. https://doi.org/10.1109/ACCESS.2019.2929211.

[13] A. G. Jivani. A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, 2(6):1930–1938, 2011. https://kenbenoit.net/assets/courses/tcd2014qta/readings/Jivani_ijcta2011020632.pdf.

[14] N. A. Khodeir. Bi-GRU Urgent Classification for MOOC Discussion Forums Based on BERT. *IEEE Access*, 9:58243–58255, 2021. https://doi.org/10.1109/ACCESS.2021.3072734.

[15] G. Kurtz, O. Kopolovich, E. Segev, L. Sahar-Inbar, L. Gal, and R. Hammer. Impact of an Instructor's Personalized Email Intervention on Completion Rates in a Massive Open Online Course (MOOC). *Electronic Journal of E-Learning*, 20(3):325–335, 2022. https://eric.ed.gov/?id=EJ1344977.

[16] E. Larson, J. Aroz, and E. Nordin. The Goldilocks Paradox: The Need for Instructor Presence but Not Too Much in an Online Discussion Forum. *Journal of Instructional Research*, 8(2):22–33, 2019. https://eric.ed.gov/?id=EJ1242593.

[17] A. Ntourmas, Y. Dimitriadis, S. Daskalaki, and N. Avouris. Assessing Learner Facilitation in MOOC Forums: A Mixed-Methods Evaluation Study. *IEEE Transactions on Learning Technologies*, 15(2):265–278, 2022. https://doi.org/10.1109/TLT.2022.3166389.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. http://jmlr.org/papers/v12/pedregosa11a.html.

[19] C. Schuster. A note on the interpretation of weighted kappa and its relations to other rater agreement statistics for metric scales. *Educational and Psychological Measurement*, 64(2):243–253, 2004. https://doi.org/10.1177/0013164403260197.

[20] L. Sha, M. Raković, Y. Li, A. Whitelock-Wainwright, D. Carroll, D. Gašević, and G. Chen. Which Hammer Should I Use? A Systematic Evaluation of Approaches for Classifying Educational Forum Posts. *International Educational Data Mining Society*, 2021. https://eric.ed.gov/?q=ED615664&id=ED615664.

[21] L. Sha, M. Raković, J. Lin, Q. Guan, A. Whitelock-Wainwright, D. Gašević, and G. Chen. Is the Latest the Greatest? A Comparative Study of Automatic Approaches for Classifying Educational Forum Posts. *IEEE Transactions on Learning Technologies*, XX:1–14, 2022. https://doi.org/10.1109/TLT.2022.3227013.

[22] W. Wang, Y. Zhao, Y. J. Wu, and M. Goh. Factors of dropout from MOOCs: a bibliometric review. *Library Hi Tech*, 2022. https://doi.org/10.1108/LHT-06-2022-0306.

[23] A. F. Wise, Y. Cui, W. Jin, and J. Vytasek. Mining for gold: Identifying content-related MOOC discussion threads across domains through linguistic modeling. *The Internet and Higher Education*, 32:11–28, 2017. https://doi.org/10.1016/j.iheduc.2016.08.001.

[24] J. Yu, L. Alrajhi, A. Harit, Z. Sun, A. I. Cristea, and L. Shi. Exploring Bayesian Deep Learning for Urgent Instructor Intervention Need in MOOC Forums. In A. I. Cristea and C. Troussas, editors, *Intelligent Tutoring Systems*, pages 78–90, Cham, 2021. Springer International Publishing. https://doi.org/10.1007/978-3-030-80421-3_10.

[25] C. Zhang, H. Chen, and C. W. Phang. Role of instructors' forum interactions with students in promoting MOOC continuance. *Journal of Global Information Management (JGIM)*, 26(3):105–120, 2018. https://doi.org/10.4018/JGIM.2018070108.

Optimizing Parameters for Accurate Position Data Mining in Diverse Classrooms Layouts

Tianze Shou
Carnegie Mellon University
tshou@andrew.cmu.edu

Conrad Borchers
Carnegie Mellon University
cborcher@cs.cmu.edu

Shamya Karumbaiah
University of
Wisconsin-Madison
shamya.karumbaiah@wisc.edu

Vincent Aleven
Carnegie Mellon University
aleven@cs.cmu.edu

ABSTRACT

Spatial analytics receive increased attention in educational data mining. A critical issue in stop detection (i.e., the automatic extraction of timestamped and located stops in the movement of individuals) is a lack of validation of stop accuracy to represent phenomena of interest. Next to a radius that an actor does not exceed for a certain duration to establish a stop, this study presents a reproducible procedure to optimize a range parameter for K-12 classrooms where students sitting within a certain vicinity of an inferred stop are tagged as being visited. This extension is motivated by adapting parameters to infer teacher visits (i.e., on-task and off-task conversations between the teacher and one or more students) in an intelligent tutoring system classroom with a dense layout. We evaluate the accuracy of our algorithm and highlight a tradeoff between precision and recall in teacher visit detection, which favors recall. We recommend that future research adjust their parameter search based on stop detection precision thresholds. This adjustment led to better cross-validation accuracy than maximizing parameters for an average of precision and recall ($F_1 = 0.18$ compared to 0.09). As stop sample size shrinks with higher precision cutoffs, thresholds can be informed by ensuring sufficient statistical power in offline analyses. We share avenues for future research to refine our procedure further. Detecting teacher visits may benefit from additional spatial features (e.g., teacher movement trajectory) and can facilitate studying the interplay of teacher behavior and student learning.

Keywords

stop detection, hyperparameters, optimization, spatial analytics, position mining, classroom analytics, position sensing

1. INTRODUCTION

The increasing accessibility and affordability of position sensing devices have fostered the application of position analytics in educational data mining [4, 18, 27, 28]. Features mined from these novel data streams have various applications, including healthcare worker training [3], the study of teaching strategies [15], and instructor dashboards [4, 18].

One key feature derived from position data is teacher or student stops in the classroom, extracted via decision rules or algorithms for *stop detection* [16, 25, 26]. For our purposes, we define stop detection as extracting timestamped and located stops (i.e., pauses in movement) from raw data that captures the movement of individuals in classrooms as a time-series of x-y coordinates. Stop detection defines a set of parameters that determine the presence of a stop or interaction between two individuals. Typically, a *radius* parameter determines a range of motion the actor does not exceed, and a *duration* parameter designates the minimal amount of time the actor is required to stay in that range of motion.

Stop detection has various nascent applications in educational data mining. Martínez-Maldonado et al. [17] used heatmaps to infer the distribution of teacher visits at different groups of students and inferred teacher strategies by investigating sequences of teacher visits targets. Similarly, An et al. [1] used dandelion diagrams (i.e., a triangular "spotlight" shape) to visualize teachers' spatial trajectory for teacher reflection tools. Other studies highlighted the importance of spatial teacher attention for learning. One study related teacher-student interactions to improved learning and engagement in a higher education physics lab [21].

With many of these applications emerging, a critical issue in stop detection is a lack of validation of the accuracy of stops to represent phenomena of interest (e.g., teacher-student interactions). Past studies made ad hoc choices for parameters used in stop detection without validating their choices of *radius* and *duration* parameters [16, 25, 26]. This is important because a lack of validation in the detection of spatial features can result in noisy variables that either do not relate to learning outcomes of interest (e.g., learning gain differences based on the frequency of teacher visits of students) or, in the worst case, lead to a biased inference. Relatedly, parameter choices need to generalize to diverse classroom settings and layouts adequately, given that the spatial movement of teachers (and the resulting distance parameters during interactions) likely vary across classroom settings and pedagogies

T. Shou, C. Borchers, S. Karumbaiah, and V. Aleven. Optimizing parameters for accurate position data mining in diverse classrooms layouts. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 310–317, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115685

[13]. For example, classrooms with technology-based learning have been reported to include spatial movements and behaviors of teachers different from more traditional classroom settings [11].

Taken together, applying prior stop detection procedures to infer teacher visits at particular students in dense K-12 classrooms requires adjustments. The current study presents a reproducible procedure to optimize stop detection parameters for K-12 classrooms. This extension is motivated by adapting parameters to infer teacher visits to students working with an intelligent tutoring system. This study reports initial baselines for detecting and validating inferred teacher visits in K-12 classrooms. Our approach includes optimizing stop detection parameters based on training data of field observations, drawing from studies outside of education that used machine learning for stop detection from video motions [9]. To achieve this, we extend an established stop detection algorithm described in Martínez-Maldonado et al. [16] to account for dense classroom layouts. Finally, we contribute guidelines regarding handling tradeoffs in teacher visit detection accuracy, namely accuracy and recall, concerning sample size. We share reproducible analysis code that includes our stop detection algorithm and its parameter tuning, including synthetic training data.[1]

2. RELATED WORK

2.1 Stop detection

One key application of stop detection in educational data mining featured in this study is to map the stops of teachers to visits of particular students in the classroom (referred to as teacher visits). We survey prior research on teacher visits and stop detection in educational data mining.

Teacher visits can relate to various constructs relevant to teaching strategies. First, teacher visits can relate to helping students. VanLehn et al. [23] developed a data-driven classroom orchestration tool to recommend teacher visits to students working with intelligent tutoring systems and make visible the limited resources of teachers to visit all students that would require help through qualitative coding of teacher-student interactions. Second, teacher visits can also relate to teacher information seeking [22] and student relationship building [12]. Given teachers' time and resource constraints to pay spatial attention to all students' needs in the classroom, past work has argued that improved learning through teacher-facing tools is partially due to improved teacher sensing and attention allocation decisions in the classroom [8]. In line with this reasoning, recent research found student idleness decreased after teacher visits when working with AI tutors [10].

Past methodological choices in stop detection algorithms have been heuristic, ad hoc, and varied. This is important as established machine learning techniques for stop detection are largely based on GPS data (cf. [19]) which do not provide the spatial granularity necessary for stop detection in classroom settings. Martínez-Maldonado et al. [16] used a distance from the teacher's x-y coordinates of 1 m to detect stops based on a heuristic of individuals' reported personal space during interpersonal interactions [20] and an ad hoc duration of the proximity of 10 s. Similarly, Yan et al. [26] classified teacher-student interactions by spatial proximity of less than 1 m for longer than 10 s. Yan et al. [25] used heuristics to determine distance thresholds between students and teachers to detect social interactions based on [6]. The distances were classified into intimate (0 – 0.46 m), personal (0.46 – 1.22 m), social (1.22 – 2.10 m), and public (2.10 m and above). Yan et al. [25] acknowledge that further validation work is desirable to assess social interactions through triangulations with more data sources.

2.2 Applications of spatial analytics in educational data mining

We identify three common use cases of spatial analytics in educational data mining. First, spatial analytics can be used to derive features for learning outcome inference. Yan et al. [28] used position data of healthcare students to assess tasks and collaboration performance in simulation-based learning and demonstrate the feasibility of using these analytics to distinguish between different levels of student performance. Yan et al. [26] used Markov chains of student interaction sequences with student and teacher as well as individualized studying primary school to model learning over eight weeks and demonstrate the feasibility of these analytics to detect low-progress students. Second, spatial analytics can guide teacher reflection and strategy. Yan et al. [27] engineered features from teacher position logs to encode proactive or passive teacher interactions. They also demonstrate the feasibility of linking these spatial analytics to different classroom spaces relating to different pedagogies [13]. Third, spatial analytics can inform instructor dashboards and in-the-moment teaching support. Fernandez-Nieto et al. [4] used epistemic network analysis to enact student movements for instructors in nursing education. They find that these enactments were consistently interpreted across multiple instructors. Similarly, Saquib et al. [18] demonstrate that position sensors worn in students' shoes in early-childhood classrooms can help teachers better plan individualized curriculums and identify student interaction needs.

2.3 The present study

Methodological choices in stop detection have mainly relied on heuristics and ad hoc decisions. Given the increasing use of stop detection and spatial analytics in educational data mining, there is a need to adapt stop detection to different classroom contexts concerning their size, spatial layout, and teaching context. Addressing this gap, this study follows three steps. First, we describe an extended algorithm for stop detection to infer teacher visits based on Martínez-Maldonado et al. [16] to account for dense classroom seating layouts in which teacher visits can relate to multiple students simultaneously. Second, we describe a reproducible procedure to optimize the parameters of that stop detection algorithm given human-coded ground-truth observations in a K-12 classroom working with an intelligent tutoring system. Third, we evaluate the accuracy of our algorithm given different thresholds for the precision of stop detection and discuss the challenges and affordances of our procedure concerning research aims and future work.

[1] github.com/Sho-Shoo/stop-detection-optimization-edm23

3. METHODS

3.1 Data

We collected training data for our stop detection algorithm on eighty-five 7th graders and one teacher in a public school in the United States, where we have obtained IRB (i.e., ethics board) approval for data collection. The data included 1) the teacher's position in the classroom, 2) classroom observation in five distinct classes across three days, and 3) student seating coordinates in the classroom, which were constant throughout the study. Each class held one session daily, and all sessions focused on algebraic equation solving. Figure 1 is a visual of the data collection site.

During all classroom sessions, students worked with an AI tutor, Lynnette [14, 24]. Lynnette is an intelligent tutoring system specialized in equations solving practice for K-12. During practice, the teacher moved around the classroom to support students. According to our classroom observations, students sometimes raised their hands and proactively asked for the teacher's attention when Lynnette's hints were insufficient.

To gather teacher position data, we used Pozyx. Pozyx is a positioning system that provides real-time location information based on automated sensing. We placed six anchors as a 2 x 3 matrix in the four borders of the classroom. All timestamped position coordinates, recorded at a one-second sampling rate, included X and Y coordinates in a 2D plane representing the classroom. Tracking tags were used to measure the coordinates of all the major objects in the classroom, including each student's desk, teacher's desk, blackboard, window, and door. These reference points were used to track teacher positions concerning students and relevant objects in the classroom.

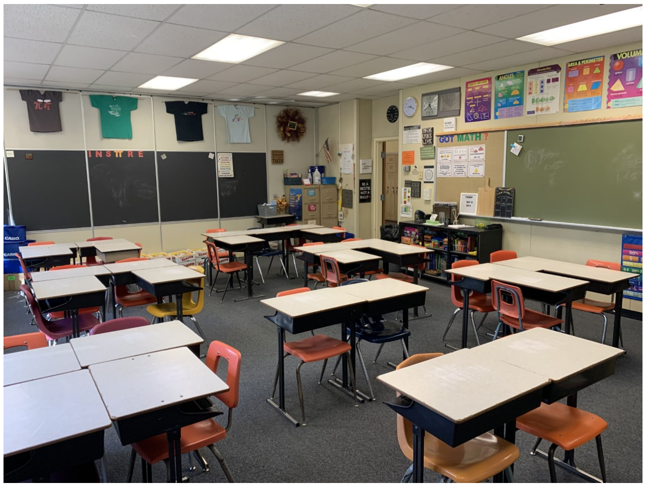


Figure 1: Middle school classroom with desks and chairs

Following procedures described in Holstein et al. [7], one observer took notes at the back of the classroom during data collection. The observer recorded teacher actions, including "monitoring class" and "helping student #1" and took notes of students' behaviors like "raising hand". The observer also noted which student a teacher interacted with. All observations were logged in real-time with time stamps using the "Look Who's Talking" software. Activities logged as on-task and off-task conversations with students or groups of students (referred to as teacher visits) served as training data for stop detection.

Table 1: Example data table for position data, observation log, and Stop Detection Output, including timestamps (t) and students (S).

	Pozyx		Observation		Prediction	
t	X	Y	Visit	Subject(s)	Stop	Inference
0	100	100	True	S3	False	NA
1	110	90	True	S3	True	S3, S1
2	200	250	False	NA	True	S3, S1
3	1000	1000	True	S1	False	NA
4	1700	1500	False	NA	True	S10

3.2 Stop detection setup and algorithm

To generate accuracy measures for our stop detection algorithm, we match human observations of the teacher visiting particular student(s) to X-Y coordinates of teachers. These timestamped observation logs of teacher visits to particular students(s) serve as ground-truth for algorithm training. We then create estimates of teacher visits based on teacher X-Y coordinates and compare these to the ground-truth stops. Notice that both observer-generated and stop detection-generated teacher visits are accompanied by student subject(s), which can relate to multiple students simultaneously. A preview of the data set for stop detection algorithm optimization is in Table 1.

Martínez-Maldonado et al. [16] proposed a stop detection algorithm based on *duration* and *radius*. The algorithm iterates through the teacher's position coordinates. A stop is established if the teacher's X-Y coordinates are within a circle defined by *radius* for a pre-defined time (*duration*). Extending on this stop detection algorithm, we propose a new method to identify the student(s) visited by the teacher during a teacher's stop. This extension is motivated by more dense classroom layouts in K-12 classrooms (including the classroom of our data collection), where students usually sit in groups, and the teacher may stand close to and interact with multiple students simultaneously. We define another parameter called "range". At the time of the stop, students seated within a circle with radius $r = range$ of the inferred stop are added as subjects of that particular teacher visit. Algorithm 1 describes the algorithm's implementation.

Our implementation of stop detection via required proximity over a minimal duration features a moving window bounded by two timestamps, t_l and t_r. The two boundaries move according to the following rules:

- If the coordinates within the time window are within a certain radius distance relative to a point coordinate, the right-side boundary t_r will increase by one second;
- Otherwise, and if $t_r - t_l \geq duration$, the interval $[t_l, t_r]$ will be denoted as a teacher visits; the visit's corresponding coordinate centroid will also be stored; and t_l will be updated to be $t_l \leftarrow t_r$;

```
Data: Teacher position data: X-Y coordinates with
      timestamp; given duration, radius, and range
      parameters
Input: $arr, result$
Output: $result$
Result: Teacher Visit Intervals
$t_l \leftarrow 0$;
$stops \leftarrow$ List();
while $t < t_{final}$ do
    $t_r \leftarrow t_l + 1$;
    while WithinRadius($position_{[t_l, t_r]}, radius$) do
        $t_r + +$;
    end
    if $t_r - t_l < duration$ then
        $t_l + +$;
    else
        $studSet =$ NearbyStuds($(t_l, t_r)$, $range$);
        $stops$.append($(t_l, t_r, studSet)$);
        $t_l \leftarrow t_r$;
end
return $stops$;
```

Algorithm 1: Stop detection algorithm proposed in this study

- If $t_r - t_l < duration$, let t_l increment by one second and continue.

3.3 Cross-validation method

Cross-validation is employed to investigate the robustness and generalizability of our stop detection algorithm. Since the dataset contains five class periods (see Section 3.1), splitting the training data into five folds and by class period is natural. There are two reasons behind this decision. First, random student splits may cause data leakage, as the stop detection error on students in the vicinity is expected to be correlated. Second, creating folds based on class period puts our algorithm to the test of accounting for differences in teacher behavior across periods. For example, one of the periods is an honors class, where students' academic performance is high.

Period-level cross-validation is conducted in five steps. First, the dataset is split into five folds by class period. Second, we define fold #n as including period #n's data as testing set and other periods' data as training set. Third, a full parameter sweep (see Section 4.1) is conducted on each fold's training data. Fourth, the best-performing parameters is selected for each fold based on the evaluation metrics described in the Evaluation Section. Fifth, the selected parameters are evaluated on each fold's test set, and evaluation metrics (precision, recall, and F_1 score) are reported.

4. PROPOSED OPTIMIZATION PROCEDURE

To better quantify the values of the three parameters, *duration*, *radius*, and *range*, in more diverse classroom layouts, we present a novel parameter search algorithm based on grid search to optimize the stop detection algorithm.

Grid search takes a pre-defined parameter search space and evaluates each parameter candidate to find a global optimum. The relatively small size of our position data ($N =$ 19,073 recorded teacher position records) also enables us to run a grid search within a reasonable time, not requiring complex optimization procedures such as gradient descent. We define the search space of the three parameters via lower and upper bounds, including a step size. The step size designates the value by which the lower bound is incremented for trialing the next parameter value until the upper bound is reached. The search bounds for range and radius were based on estimations of the minimal and maximal distance between students observed in our classroom layout. In addition, the minimal duration was based on the coder's experience of teacher movement in the classroom, which would entail brief stops for spatial orientation of below 3 s. In contrast, the maximum duration was based on not exploring minimal durations three times as long as those used in prior work (cf. [16]), which we deemed not meaningful. Our search space was *duration*: $[3, 30)$ where the step size is three and the unit is second, *radius*: $[200, 2000)$ where the step size is 200 and the unit is millimeter, and *range*: $[100, 2000)$ where the step size is 200 and the unit is millimeter.

To compare teacher visit detection accuracy across different parameter combinations, we define three metrics for evaluation: hits, misses, and false alarms. We further describe these measures in the next section.

4.1 Evaluation

By treating observation logs as the ground truth, we introduce a function, `Evaluate`, that outputs three metrics, hits, misses, and false alarms, to describe the alignment between these ground truth representations of teacher visits and the inferred subjects of our stop detection algorithm.

Suppose an arbitrary teacher visit documented in the observation log is v_i, and its corresponding timestamp is t_i. During v_i, a set of student subjects, S_i, were visited. S_i is the ground truth subject set corresponding to ground truth visit v_i. For each ground-truth observation, compute a timeframe between time stamp $[t_i - 5, t_i + 5]$, which is a 10-second window. We are examining a time frame instead of a single time point because classroom observations of teacher visits include a natural degree of imprecision. The human coder described their time stamp recording of teacher visits and the time of the actual visit to differ by up to 10 s, the size of or timeframe window. In other words, the observation record may be entered a few seconds earlier or later than the true starting time of an event, with ±5 seconds being a reasonable estimate as reported by the observer. By filtering variables `posStops` and `inferredSubj` with only entries in time frame $[t_i - 5, t_i + 5]$, we can obtain an inferred subject set G_i. We define $hit_i = |S_i \cap G_i|$, $miss_i = |S_i \setminus G_i|$, and $falseAlarm_i = |G_i \setminus S_i|$.

A hit is an element S_i and G_i have in common: a correctly inferred student subject that was stopped at. Miss counts the true subjects our stop detection fails to capture, while false alarm keeps records of incorrect subjects tagged by stop detection.

Recall all calculations are based on iterating through the observation log while gathering algorithm-extracted teacher visits within time frame $\bigcup_i [t_i - 5, t_i + 5]$ with i being in stop index in the observation log. This does not account for incorrectly inferred teacher visits which were never gathered within time frames. We call this collection of unchecked detected stops V. Suppose an arbitrary visit in V is v_j, and its corresponding inferred subject set G_j. We can also treat these inferences as false alarms: $falseAlarm_j = |G_j|$. Notice this subscript j is different from the previous i. We call these false alarm counts to be "outside" since they are outside the unionized time frames. Conversely, $falseAlarm_i$ represents "inside" false alarms. We evaluate all algorithms based on false alarms inside and outside designated time frames. Still, we note that for some applications, an evaluation of inside false alarms only might be more desirable. To evaluate algorithmic accuracy, we sum the total number of hits, misses, and false alarms for a given parameter combination. We introduce measures that combine these three metrics for optimization, namely precision and recall, which are analogous to precision and recall in machine learning classification tasks:

$$recall = \frac{hit}{hit + miss} \tag{1}$$

$$precision = \frac{hit}{hit + falseAlarm} \tag{2}$$

While precision designates the probability of an inferred teacher visit to be correct according to observation logs, recall is the probability of any given observation log teacher visit to be detected via stop detection. We select optimal parameter combinations for stop detection on a global maximization of precision and recall by evaluating all parameter combinations in a grid search based on our search space. For larger data, less extensive optimization algorithms, such as gradient descent, may be preferable.

The following algorithms (Algorithm 2) demonstrate how the grid search is carried out together with `Evaluate` (Algorithm 3) that implements the aforementioned set calculation:

```
Data: Teacher Position Data and Observation Logs
Result: Hits, Misses, and False Alarms for Each
        Parameter Combination
for $d$ in $durationGrid$ do
    for $r$ in $radiusGrid$ do
        $posStops \leftarrow$ GetStops($teacherPos$) ;
        $obsStops \leftarrow$ GetStops($obsLog$);
        for $rng$ in $rangeGrid$ do
            $inferredSubj \leftarrow$ GetSubj($posStops, rng$);
            $hit, miss, FA \leftarrow$
              Evaluate($posStops, inferredSubj, obsLog$);
            SaveToFile($hit, miss, FA$);
        end
    end
end
```

Algorithm 2: Parameter sweep algorithm

```
$seenStops \leftarrow$ List();
for $obsStop$ in $obsLog$ do
    $t \leftarrow obsStop.time$;
    $stops \leftarrow posStop[t-5, t+5]$;
    $seenStops$.append($stops$);
    $S \leftarrow$ SubjectOf($obsStop$);
    $G \leftarrow$ SubjectOf($stops$);
    $hit_i, miss_i, FA_i \leftarrow$ SetOps($S, G$);
    $hit, miss, FA \leftarrow hit + hit_i, miss + miss_i, FA + FA_i$;
end
for $stop$ in $posStops$ and $stop$ not in $seenStops$ do
    $FA_j += |$SubjectOf($stop$)$|$;
    $FA += FA_j$;
end
return $hit$, $miss$, $FA$;
```

Algorithm 3: `Evaluate` function body

5. RESULTS

5.1 Parameter sweep results

We tune stop detection algorithm parameters with respect to precision and recall. As a first step, to gauge the overall performance of our algorithm given different parameter settings, we visualize precision and recall for all of our parameter combinations in Figure 2.

Based on Figure 2, we find that recall deteriorates faster with increasing precision than precision deteriorates with increasing recall. This means that improving precision in our algorithm concerning our training data comes with a relatively high recall cost.

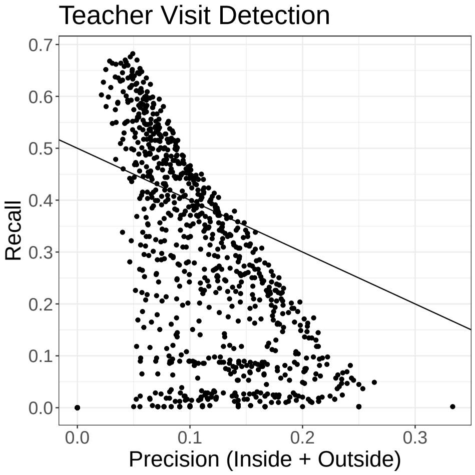


Figure 2: Scatter plot of precision and recall (inside and outside) for all parameter combinations of our stop detection algorithm evaluated against ground-truth observations of teacher visits, including a reference line with slope -1.

Based on this finding, we identify two ways of sampling an optimal set of parameters for precision and recall. The first set of parameters is derived from maximizing the average of precision and recall (referred to as "absolute maximization"). The resulting set of parameters is $duration = 6$, $radius = 1800$, and $range = 1900$. The radius and range are close to the upper bound of our search space. This may be due to the relatively fast deterioration of recall over precision, overemphasizing recall when averaging precision and recall, and leading to very liberal stop detection. Therefore, we select the second set of parameters based on a minimally required precision cutoff (referred to as "conditional maximization"). We set this cutoff to be $precision > 0.2$. This selection strategy's resulting parameters are *duration* = 21, *radius* = 600, and *range* = 700. Notice that the higher the precision cutoff, the lower the number of detected stops will be. Therefore, one way of resolving the precision-recall tradeoff is to set the precision cutoff low enough to obtain a sample size sufficiently large for a given study design. For example, if the study design includes a two-sided t-test of whether teacher visits are, on average, longer for low- than high-prior knowledge students, a sufficiently large number of stops assuming a power of $1 - \beta = 0.8$ and effect size $d = 0.3$ would be around $N = 175$ stops.

5.2 Parameter weights

We fit an ordinary least square (OLS) regression inferring the average of precision and recall to approximate the relative feature importance of our three parameters (i.e., duration, radius, and range) on teacher visit detection accuracy. To compare effect sizes, we Z-standardize all three parameters to a mean of 0 and a standard deviation of 1. We report the result of the regression in Table 2.

Table 2: OLS regression results of parameter weights on the average of precision and recall of teacher visit detection.

Predictors	β	$CI_{95\%}$	p
Intercept	0.25	0.24 – 0.25	**<0.001**
Duration	-0.02	-0.03 – -0.02	**<0.001**
Radius	0.01	0.00 – 0.01	**.003**
Range	0.06	0.05 – 0.06	**<0.001**
$R^2_{adjusted}$	42.0%		

According to Table 2, while all three parameters had a significant association with the average of precision and recall, range had the largest standardized effect size ($\beta = 0.06$, $p < .001$).

5.3 Cross-validation

Given the tradeoff between precision and recall described in Section 5.1, we report cross-validation results broken out by absolute and conditional maximization. Table 3 reports the chosen parameters by fold compared to those chosen by running parameter sweep on all position data.

Table 3: Parameters selected per CV fold.

Fold	Maximization	*duration*	*radius*	*range*
1	conditional	30	600	700
	absolute	6	1800	1900
2	conditional	18	600	700
	absolute	6	1800	1900
3	conditional	21	1200	700
	absolute	6	1800	1900
4	conditional	18	600	700
	absolute	6	1800	1900
5	conditional	21	600	700
	absolute	6	1800	1900
all	conditional	21	600	700
	absolute	6	1800	1900

The parameters chosen for each fold are comparable to the optimal parameters when fitting parameters to the full training data set. Table 4 displays the means and standard deviations of the three performance metrics (F_1, precision, and recall) across the five folds.

Table 4: CV evaluation metrics per fold.

Maximization	Metric	*M*	*SD*
Absolute	F_1 score	0.09	0.01
	Precision	0.05	0.00
	Recall	0.68	0.03
Conditional	F_1 score	0.18	0.02
	Precision	0.21	0.07
	Recall	0.17	0.05

Conditional maximization yielded an average F_1 score twice as large as absolute maximization (0.18 compared to 0.09). Moreover, absolute maximization corresponded to liberal teacher visit detection (i.e., low precision at high recall), while conditional maximization led to more balance between precision and recall.

6. DISCUSSION AND CONCLUSIONS

Stop detection and spatial analytics receive increasing attention in educational data mining. Yet, with past stop detection parameter settings being based on heuristics, there is a need to evaluate and optimize stop detection in diverse classroom settings and layouts. In this study, we extended a popular stop detection algorithm to detect teacher visits to particular student(s) in a K-12 math classroom working with intelligent tutoring systems. We introduced metrics to evaluate the algorithm's accuracy against ground truth human observations of teacher visits. Our three main findings are as follows:

First, we find a large variability in stop detection accuracy given different parameter choices. This is important as past work has primarily relied on ad hoc or heuristic parameter settings in stop detection [16, 25, 26]. As an implication for research, spatial features other than inferred teacher visits may afford similar validation work and adaptation to diverse classroom contexts as presented in this study. Potential outcomes of interest include the total time teachers spent attending to different students, the average visit duration, and the dispersion, or entropy, of visits to students [15]. Our proposed optimization procedure may be readily extended to infer these spatial features.

Second, we establish a benchmark for teacher visit detection accuracy that future research may pick up. To improve accuracy, we contribute a reproducible procedure to adapt our algorithm to diverse classroom layouts and contexts. We described strategies to weight precision and recall to derive meaningful sets of teacher visits for research. Importantly, our results indicate that setting a precision threshold during parameter fitting yields superior cross-validation accuracy. More generally, we find a precision-recall tradeoff in detecting teacher visits that favors recall over precision, as precision came with a higher cost in the tradeoff. This might be due to the nature of our data set, as our classroom layout included dense groups of students compared to previous studies using open learner spaces [16]. Teachers may have interacted only with a subset of students sitting in a group, leading to larger ranges for satisfactory recall at an excess of false positives and diminishing precision. Coding teachers' proximity to groups rather than teacher-student interactions might be a more tractable prediction task based on position coordinates alone in dense classroom layouts. For the detection of teacher visits, the high cost of precision in dense classrooms may result in lower statistical power through smaller sample sizes of resulting visits as the number of detected visits diminishes with increasing precision. Researchers may adjust precision thresholds accordingly. We note, however, that with lower precision, statistical associations might be less likely to be detected as the false positive teacher visits introduce noise to features. Therefore, we recommend future research to estimate the expected number of stops during classroom sessions ahead of the data collection to plan sample sizes accordingly. Multiplying estimates of the number of expected stops with stop detection precision could yield an estimate for the number of detected stops for power analyses.

Third, we find that range (i.e., the minimally required distance of students to the teacher during visits) had the largest association with teacher visit detection accuracy. Notice that range was the new parameter we defined to detect multiple students in proximity to the teacher during stops to account for dense classroom layouts. This suggests that the largest improvement to our algorithm might be achieved through optimizing the decision rules for tagging groups of students in proximity to the teacher. One such improvement might be approximating the teacher's orientation during the visit based on past movement trajectory. Future work may test whether excluding students not faced by the teacher (e.g., seated behind their back) from stop detection improves accuracy. Finally, further improvements of the range parameter appear desirable, particularly for dense classroom layouts with groups of students, such as K-12 classrooms.

6.1 Limitations and future work

We acknowledge limitations to our current methodology that future research may improve upon. First, limitations may emerge from how our ground truth observation data of teacher visits were coded. In particular, manual coders could only code visits to an accuracy level of a time frame of 10 s. Future work may improve training data quality by using more coders and establishing inter-rater reliability for the coding of visits or other means of automatically generating observation logs during classroom sessions, for example, by recording observations verbally with a microphone rather than typing them into a laptop. More accessible tools for coding may reduce the time lag in human coding during model optimization and improve overall stop detection accuracy. We note that the quantitative definition of stops may differ based on the research context. Hence, future work may refine coding schemes for coders to capture spatial attributes of different research contexts (e.g., coding 1-to-1 interactions between teachers and students compared to group visits). In both cases, our optimization procedure allows for adapting stop detection to such complexities for more nuanced explorations in offline analyses.

Second, our algorithm may require more sophisticated decision rules to achieve better accuracy. Based on our evaluation, inferring teacher visits may benefit from additional spatial features for algorithm training other than the teacher's spatial position only (e.g., information about the teacher's movement trajectory before a visit). Our relatively low cross-validation F_1 score of 0.18 may relate to the challenge of inferring teacher visits to particular students when the students of interest sit close to others not visited. Next to teacher visits, teacher proximity may also encode teacher attention effects on student learning, such as motivational and performance differences through mere presence [2, 5]. Future extensions of our algorithm could also consider specific teacher movement strategies. For example, models could calibrate to the usual distance of teachers when interacting with students. Teachers may have different distances from different students (e.g., due to some students sitting in the back of the classroom). Fitting a parameter to student characteristics to adjust the distance in stop detection may improve accuracy while being sufficiently generalizable to new students. Similarly, future research may also consider fitting the stop detection parameters as a function of spatial attributes instead of being static. Under a dynamic set of parameters, the detection algorithm may be better able to differentiate between teacher standing at the periphery of the classroom observing and actually visiting students in the middle of the classroom.

Third, the cross-validation indicates that our stop detection algorithm and optimization procedure are generalizable across different class periods. However, this study only explored one classroom layout setting: a dense layout with grouped seating typically found in US K-12 classrooms. More research is needed to gauge the performance of our algorithm and optimization procedure for other seating arrangements.

We see two central use cases of our proposed stop detection algorithm and optimization procedure. First, future research could use our adaptive algorithm to more accurately mine stops and investigate teachers' attention distribution at a lower cost. Our algorithm can learn relevant stop detection parameters from human-coded examples of teacher visits and automatically generate a teacher visit distribution from optimized parameters, facilitating data collection. Second, our stop detection algorithm can be incorporated into teacher-facing reflection and orchestration tools, where stop detection can serve as a feature for teacher-facing analytics. These applications can help facilitate the study of the interplay of teacher behavior and student learning.

7. REFERENCES

[1] P. An, S. Bakker, S. Ordanovski, C. L. Paffen, R. Taconis, and B. Eggen. Dandelion diagram: aggregating positioning and orientation data in the visualization of classroom proxemics. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–8, 2020.

[2] R. Bdiwi, C. de Runz, S. Faiz, and A. Ali-Cherif. Smart learning environment: Teacher's role in assessing classroom attention. *Research in Learning Technology*, 27, 2019.

[3] V. Echeverria, R. Martinez-Maldonado, T. Power, C. Hayes, and S. B. Shum. Where is the nurse? towards automatically visualising meaningful team movement in healthcare education. In *International Conference on Artificial Intelligence in Education*, pages 74–78. Springer, 2018.

[4] G. M. Fernandez-Nieto, R. Martinez-Maldonado,

K. Kitto, and S. Buckingham Shum. Modelling spatial behaviours in clinical team simulations using epistemic network analysis: methodology and teacher evaluation. In *LAK21: 11th International Learning Analytics and Knowledge Conference*, pages 386–396, 2021.

[5] B. Guerin. Mere presence effects in humans: A review. *Journal of Experimental Social Psychology*, 22(1):38–77, 1986.

[6] E. T. Hall. *The Hidden Dimension*. Anchor, 1966.

[7] K. Holstein, B. M. McLaren, and V. Aleven. Spacle: investigating learning across virtual and physical spaces using spatial replays. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, pages 358–367, 2017.

[8] K. Holstein, B. M. McLaren, and V. Aleven. Student learning benefits of a mixed-reality teacher awareness tool in ai-enhanced classrooms. In *International Conference on Artificial Intelligence in Education*, pages 154–168. Springer, 2018.

[9] Y. Jin, G. Suzuki, and H. Shioya. Detecting and visualizing stops in dance training by neural network based on velocity and acceleration. *Sensors*, 22(14):5402, 2022.

[10] S. Karumbaiah, C. Borchers, T. Shou, A.-C. Falhs, P. Liu, T. Nagashima, N. Rummel, and V. Aleven. A spatiotemporal analysis of teacher practices in supporting student learning and engagement in an AI-enabled classroom. In *AIED23: 24th International Conference on Artificial Intelligence in Education*, 2023.

[11] A. Kessler, M. Boston, and M. K. Stein. Exploring how teachers support students' mathematical learning in computer-directed learning environments. *Information and Learning Sciences*, 2019.

[12] A. Kwok. Classroom management actions of beginning urban teachers. *Urban Education*, 54(3):339–367, 2019.

[13] F. V. Lim, K. L. O'Halloran, and A. Podlasov. Spatial pedagogy: Mapping meanings in the use of classroom space. *Cambridge Journal of Education*, 42(2):235–251, 2012.

[14] Y. Long and V. Aleven. Gamification of joint student/system control over problem selection in a linear equation tutor. In S. Trausan-Matu, K. E. Boyer, M. Crosby, and K. Panourgia, editors, *Intelligent Tutoring Systems*, pages 378–387, Cham, 2014. Springer International Publishing.

[15] R. Martinez-Maldonado, V. Echeverria, J. Schulte, A. Shibani, K. Mangaroska, and S. Buckingham Shum. Moodoo: indoor positioning analytics for characterising classroom teaching. In *International Conference on Artificial Intelligence in Education*, pages 360–373. Springer, 2020.

[16] R. Martinez-Maldonado, J. Schulte, V. Echeverria, Y. Gopalan, and S. B. Shum. Where is the teacher? digital analytics for classroom proxemics. *Journal of Computer Assisted Learning*, 36(5):741–762, 2020.

[17] R. Martínez-Maldonado, L. Yan, J. Deppeler, M. Phillips, and D. Gašević. Classroom analytics: Telling stories about learning spaces using sensor data. In *Hybrid Learning Spaces*, pages 185–203. Springer, 2022.

[18] N. Saquib, A. Bose, D. George, and S. Kamvar. Sensei: sensing educational interaction. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4):1–27, 2018.

[19] V. Servizi, F. Pereira, M. Anderson, and O. Nielsen. Mining user behaviour from smartphone data, a literature review. 12 2019.

[20] M. Sousa, D. Mendes, D. Medeiros, A. Ferreira, J. M. Pereira, and J. Jorge. Remote proxemics. In *Collaboration Meets Interactive Spaces*, pages 47–73. Springer, 2016.

[21] J. B. Stang and I. Roll. Interactions between teaching assistants and students boost engagement in physics labs. *Physical Review Special Topics-Physics Education Research*, 10(2):020117, 2014.

[22] A. van Leeuwen, N. Rummel, and T. Van Gog. What information should cscl teacher dashboards provide to help teachers interpret cscl situations? *International Journal of Computer-Supported Collaborative Learning*, 14(3):261–289, 2019.

[23] K. VanLehn, H. Burkhardt, S. Cheema, S. Kang, D. Pead, A. Schoenfeld, and J. Wetzel. Can an orchestration system increase collaborative, productive struggle in teaching-by-eliciting classrooms? *Interactive Learning Environments*, 29(6):987–1005, 2021.

[24] M. Waalkens, V. Aleven, and N. Taatgen. Does supporting multiple student strategies lead to greater learning and motivation? investigating a source of complexity in the architecture of intelligent tutoring systems. *Computers & Education*, 60(1):159–171, 2013.

[25] L. Yan, R. Martinez-Maldonado, B. G. Cordoba, J. Deppeler, D. Corrigan, G. F. Nieto, and D. Gasevic. Footprints at school: Modelling in-class social dynamics from students' physical positioning traces. In *LAK21: 11th International Conference on Learning Analytics and Knowledge*, pages 43–54, 2021.

[26] L. Yan, R. Martinez-Maldonado, B. Gallo Cordoba, J. Deppeler, D. Corrigan, and D. Gašević. Mapping from proximity traces to socio-spatial behaviours and student progression at the school. *British Journal of Educational Technology*, 2022.

[27] L. Yan, R. Martinez-Maldonado, L. Zhao, J. Deppeler, D. Corrigan, and D. Gasevic. How do teachers use open learning spaces? mapping from teachers' socio-spatial data to spatial pedagogy. In *LAK22: 12th International Learning Analytics and Knowledge Conference*, pages 87–97, 2022.

[28] L. Yan, R. Martinez-Maldonado, L. Zhao, S. Dix, H. Jaggard, R. Wotherspoon, X. Li, and D. Gašević. The role of indoor positioning analytics in assessment of simulation-based learning. *British Journal of Educational Technology*, 2022.

Effective Evaluation of Online Learning Interventions with Surrogate Measures

Ethan Prihar
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA, USA
ebprihar@wpi.edu

Kirk Vanacore
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA, USA
kpvanacore@wpi.edu

Adam Sales
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA, USA
asales@wpi.edu

Neil Heffernan
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA, USA
nth@wpi.edu

ABSTRACT

There is a growing need to empirically evaluate the quality of online instructional interventions at scale. In response, some online learning platforms have begun to implement rapid A/B testing of instructional interventions. In these scenarios, students participate in series of randomized experiments that evaluate problem-level interventions in quick succession, which makes it difficult to discern the effect of any particular intervention on their learning. Therefore, distal measures of learning such as posttests may not provide a clear understanding of which interventions are effective, which can lead to slow adoption of new instructional methods. To help discern the effectiveness of instructional interventions, this work uses data from 26,060 clickstream sequences of students across 31 different online educational experiments exploring 51 different research questions and the students' posttest scores to create and analyze different proximal surrogate measures of learning that can be used at the problem level. Through feature engineering and deep learning approaches, next-problem correctness was determined to be the best surrogate measure. As more data from online educational experiments are collected, model based surrogate measures can be improved, but for now, next-problem correctness is an empirically effective proximal surrogate measure of learning for analyzing rapid problem-level experiments. The data and code used in this work can be found at https://osf.io/uj48v/.

Keywords

Surrogate Measures, Measures of Learning, A/B Testing, Educational Experiments

1. INTRODUCTION

There is a growing need to empirically evaluate the quality of online instructional interventions at scale. This is in part motivated by the lack of empirical evidence for many existing interventions, especially in mathematics. According to Evidence for ESSA, a website that tracks empirical research on educational practices created by the Center for Research and Reform in Education at Johns Hopkins University School of Education, only four technology based interventions have strong evidence for improving students' mathematics skills [4]. In response, more and more online learning platforms are creating infrastructure to run randomized controlled experiments within their platforms [19, 11, 18] in order to increase the impact of the their programs on student learning and facilitate research in the field. This infrastructure allows for rapid A/B testing of different instructional interventions. In an A/B testing scenario, students assigned to particular assignments or problems within these online learning platforms will be automatically randomized to one of multiple experimental conditions in which different instructional interventions will be provided to them. While this paradigm allows for rapid testing of many hypotheses, this rapid testing environment makes statistical analysis difficult. In some cases, students participate in many randomized controlled experiments in parallel or in quick succession. For example, in ASSISTments, an online learning platform in which students complete pre-college level mathematics assignments [8], students can be randomized between different instructional interventions for each mathematics problem in their assignment. In these scenarios, it is important to evaluate the effect of the interventions as quickly as possible. If one were to wait until the end of a section of the curriculum, or even the end of the current assignment before evaluating students' mastery of the subject matter, then the effect of an intervention for a single problem near the beginning of the assignment would be obfuscated by the effects of all the following interventions. For this reason, prior work has only used students' behavior on the problem they attempted after receiving an intervention but before receiving another intervention to evaluate the effectiveness of the first intervention [12, 16]. However, the measures used in prior work were chosen based on theory, without any empirical evidence that they are in fact an effective surrogate measure of learning.

E. Prihar, K. Vanacore, A. Sales, and N. Heffernan. Effective evaluation of online learning interventions with surrogate measures. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 318–324, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115703

To address the lack of empirical evidence for these proximal surrogate measures of learning, the first goal of this work was to create a variety of surrogate measures from students' clickstream data on the problem they attempted after receiving an experimental intervention. These measures were created through feature engineering, discussed in Section 3, and model fitting, discussed in Sections 4.1 and 4.2.

After creating surrogate measures, The second goal of this work was to evaluate how effective these measures were at estimating the treatment effects between pairs of conditions in online experiments. To achieve this goal, data was collected to compare 51 different pairs of conditions from 31 assignment-level online experiments with posttests in which students were exposed to the same intervention multiple times within the same assignment, but were not exposed to any other interventions. By determining the extent to which each measure was a surrogate for students' posttest scores, discussed more in Sections 2.3 and 4.4, the surrogate measures could be compared to each other.

To summarise, this work strives to answer the following two research questions:

1. What surrogate measures can be created from short sequences of students' clickstream data?
2. Which of these surrogate measures is the best surrogate for posttest score?

2. BACKGROUND

2.1 Rapid Online Educational Experimentation

Experimentation is a cornerstone of formative improvement of online instructional interventions [18, 1]. Systems like ASSISTments E-TRIALS were established to allow researchers to test learning theories and feature ideas through experiments within online mathematics assignments [11]. Using systems like E-TRIALS, students are randomized between different assignment-level interventions and complete a posttest at the end of their assignment to evaluate their learning.

Although assignment-level experiments provide some relevant information to online program designers, these designers are faced with a nearly infinite number of decisions about what features to build and how to build them. Since only one causal inference can be estimated from each manipulation [9], designing assignment-level experiments for each potentially impactful variant of a feature is often infeasible. Rapid online educational experimentation provides a more efficient alternative to more traditional assignment-level experiments by assigning students to a condition at each problem and instead of requiring students to complete a posttest, using the student's performance on the subsequent problem as the outcome.

One example of rapid online educational experimentation is the TeacherASSIST system, which randomizes students between crowdsourced hints and explanations [12]. In this system, there were over 7,000 support messages produced by 11 educators [16]. Each time a student attempted a problem for which they were provided with a randomly selected support message, their subsequent problem was used to evaluate the quality of the support. This system allowed for a much more efficient deployment of experiments and evaluation of feature nuances.

2.2 Unconfounded Outcomes For Rapid Online Experiments

In order for rapid online experimentation to lead to causal inference, we must identify outcomes that are unconfounded by the other experimental manipulations to which a student was exposed. Distal outcomes, such as end-of-unit or assignment-level posttest scores, do not allow a researcher to determine which of the treatments the student was exposed to during the experiment produced the effect. An alternative, used by [12, 16] to evaluate TeacherASSIST, is to use data from the problem students completed directly after the experimental condition, i.e., next-problem measures.

Although individual students' behaviors and performance may be influenced by the aggregate of experimental manipulations within an assignment, the average difference in next-problem measures is unconfounded due to the random assignment at the problem level. Next-problem measures are unconfounded by either the prior experimental conditions or next-problem experimental conditions because the assignment to each condition is independently random and therefore the effects of the prior and post-conditions are zero. Therefore, the remaining difference in the next-problem measures between treatment and control is an unconfounded measure of the treatment effect.

2.3 Surrogate Measures

Although measures taken during the next problem after the experiment, such as next-problem correctness, are unconfounded by other experiments within the problem set, it is not yet known whether these measure are good estimates of distal outcomes. In assignment-level A/B testing, a researcher creates a posttest designed to measure the expected effect of the treatment condition compared to the control condition, but within online instructional interventions, the next problem was designed for pedagogical purposes, not to evaluate the effects of the intervention. Therefore, to use next-problem measures to validate the impact of a condition, we must validate whether these measures assess researchers' outcomes of concern.

One way to think about these next-problem measures is as surrogate measures. Surrogate measures are used in medical experiments when the outcome is either difficult to assess or distal [17]. Surrogates can either have causal or correlation relations to the outcome [10]. Validating causal surrogates requires a causal path from the treatment to the surrogate and subsequently to the outcome, such that the indirect path through the surrogate has a larger effect than the direct path through from the treatment to the outcome. Alternatively, an associative surrogate is valid when the following three criteria are met [10]:

1. There is a monotonic relationship between the treatment effect on the surrogate and the treatment effect on the outcome across experiments.

2. When the treatment effect on the surrogate is zero, the treatment effect on the outcome is also zero.
3. The treatment effect on the surrogate predicts the treatment effect on the outcome.

In this work, various next-problem measures are evaluated for their effectiveness as an associative surrogate measure of posttest scores.

3. DATA AGGREGATION

3.1 Data Source

The data used in this work comes from ASSISTments, an online learning platform that focuses on pre-college mathematics curricula. In July, 2022 ASSISTments released a dataset of 88 randomized controlled experiments that were conducted within the platform since 2018 [?]. These experiments compared various assignment-level and problem-level interventions. For example, in one experiment, students were randomized between receiving either open response problems, or multiple choice problems during and assignment, then their learning was measured using a posttest.

In this work, the experimental assignments from ASSISTments that had posttests were used in order to compare learning measures derived from a student's clickstream data on the problem immediately after receiving an intervention for the first time to their posttest score. To avoid bias from missing posttest scores, only data from experiments in which there was no statistically significant difference in students' completion rates between conditions were used, and students that did not complete the posttest were excluded from the analysis. In some contexts it would be better to impute missing posttest scores as the minimum score. However, the purpose of this work was to create a surrogate measure for posttest score in situations where it is infeasible to require students to complete a posttest, and therefore it seems more appropriate to remove missing posttest scores to ensure that the surrogate measures students' posttest scores, not their propensity to complete an assignment. This additional filtering step removed only one of the ASSISTments experiments from the analysis. Additionally, the data used in this work is limited to students who participated in the experiments prior to July 23rd, 2021. On July 23rd, 2021 all unlisted YouTube videos created prior to 2017 were made private [6]. Many of the experiments included YouTube videos uploaded prior to 2017, which were made private, ruining the experiments that contained them. In total, 26,060 clickstream sequences of a student completing a problem and their corresponding posttest score were collected for model training and analysis across 51 different research questions within 31 different experimental assignments. These sequences and the code used to evaluate them has been made publicly available and can be found at https://osf.io/uj48v/.

3.2 Expert Features

As established by prior work, i.e. ([12, 16, 14]), collecting data to evaluate the effectiveness of an intervention is often limited to data from the next problem in a student's assignment before they receive another intervention. This work extracted five expert features from students' clickstream data on their next problem that have been useful predictors of student behavior in prior work [20, 21]. Table 1 describes the expert features evaluated for their effectiveness as a surrogate measure of posttest score.

3.3 Clickstream Data

In addition to expert features, this work used deep learning to create surrogate measures of learning from students' clickstream data. The clickstream data consisted of the action sequences of students within the ASSISTments tutor from the time they start the problem after they received an experimental intervention to the time they either receive another intervention or complete the problem. This short window of time is not confounded by other experimental interventions and is likely to give the clearest insight into the impact of experimental interventions being tested in quick succession.

The students' clickstream data was broken down into a series of one-hot encoded actions followed by the time since taking the last action. The first action was always ”problem_started”, therefore this action was dropped from students' clickstreams prior to being given to a deep learning model. The time since taking the last action was log-transformed in order to weight the difference between short time periods more than long time periods and to reduce the impact of large outliers, which are due to students walking away from their computers during assignments and returning later. Additionally, the log-transformed times are scaled within the range [0, 1]. Scaling the time within the same range as the one-hot encoded actions helps the model balance the importance of the different features. Each action sequence was equal in length to the longest action sequence, which was 12 actions. When students took less than the maximum number of actions, their action sequences were zero padded from the start of the sequence. Table 2 provides an example sequence of a student's clickstream data in which a student unsuccessfully attempted to get a problem correct twice, then took a break, then returned to their assignment, got the problem incorrect again, and then on their fourth attempt, got the problem correct. The first six columns contain all zeros because the student only took a total of six actions. This representation of students' clickstream action sequences was chosen because of its success in previous work for various prediction tasks [20, 15, 21].

4. METHODOLOGY

4.1 Expert Feature-Based Models

To derive a surrogate measure of learning from the expert features, three approaches were taken. The first approach was to simply use each expert feature as a surrogate measure of learning, the second approach was to fit a linear regression on posttest score using the expert features as input, and the third approach was to fit a linear regression on the treatment effect on posttest score using the treatment effects on each expert feature as input. The third was included because if the goal is to predict the treatment effect on posttest score, than it might be more effective to fit a model that combines the treatment effects on different expert features into the treatment effect on posttest score than to simply predict posttest score. This would be advantageous in a scenario where there was information in the expert features that was predictive of a student's propensity to learn independent of

Table 1: Expert Features

Feature Name	Description
Correctness	A binary indicator of whether or not the student answered the problem correctly on their first try without tutoring of any kind.
Tutoring Requested	A binary indicator of whether or not the student requested tutoring of any kind.
No Attempts Taken	A binary indicator of whether or not the student did not make any attempts to answer the problem.
Attempt Count	The number of attempts made by the student to answer the problem.
First Response Time	The natural log of the total seconds from when the problem was started to when the student submitted an answer or requested tutoring of any kind for the first time.

Table 2: A Student's Clickstream Data Sequence After Processing

Feature Name	Clickstream Data Sequence											
problem_resumed	0	0	0	0	0	0	0	0	1	0	0	0
tutoring_requested	0	0	0	0	0	0	0	0	0	0	0	0
wrong_response	0	0	0	0	0	0	1	1	0	1	0	0
correct_response	0	0	0	0	0	0	0	0	0	0	1	0
problem_finished	0	0	0	0	0	0	0	0	0	0	0	1
time_since_last_action	0.00	0.00	0.00	0.00	0.00	0.00	0.62	0.51	6.39	0.12	0.38	0.01

the intervention they were given. In that scenario, a model trained to predict posttest score might learn to rely on that information, which would lead the model to predict more similar posttest scores between different experimental conditions than were actually observed. By directly predicting the treatment effect on posttest score, the model must learn to use the features that are predictive of the effect of the experimental conditions. The downside of this approach is that each research question's data was reduced to a single sample in the regression. Therefore, while the second approach had the full 26,060 samples of student data to fit on, the third approach only had 51 samples to fit on; one for each research question.

4.2 Deep Learning Models

Two deep learning approaches were used to create a surrogate measure of learning from students' clickstream data. Both approaches trained a recurrent neural network to predict students' posttest scores given their clickstream data using Bidirectional LSTM layers [22, 5], which read the clickstream data both forward and backward to learn the relationship between students' actions and their posttest scores. Following the same intuition as Section 4.1, the first model used the mean squared error of its posttest score predictions as its loss function, the second model used the squared error of the treatment effect calculated from its posttest score predictions as its loss function. Essentially, the first model was trained to predict accurate posttest scores, and the second model was trained to predict posttest scores that would lead to the same treatment effect estimates as the actual posttest scores.

4.3 Model Training

To fairly evaluate the surrogate measures of learning, each model was trained and evaluated using a leave-one-out cross-validation approach partitioned by the experimental assignment. Many of the experimental assignments evaluated multiple research questions using the same control. Therefore, all the research questions in the held-out experimental assignment were evaluated using the model trained on all the other experimental assignments, as opposed to performing leave-one-out cross-validation partitioned by research question. This ensured that no data was shared between the training data and the held-out data.

For the expert feature-based models, an ablation study was performed to identify which combination of features, when used as input, led to the highest correlation between surrogate measure and posttest treatment effects. In this ablation study, the models were trained first using all of the expert features as input, and then models were trained using all but one of the features. If any of the all-but-one-feature models out-performed the model with all the features, then that model became the best model so far, and more models were trained using all but one of the features in the new best model. Eventually, the best model will not have improved from removing any of its features, denoting that this model has the optimal set of features as input.

For the deep learning models, the models were initialized, trained, and evaluated ten times, averaging the results of each evaluation. Neural networks cannot be solved for the optimal value of their weights; gradient descent is instead used to optimize them starting from random initializations. These random initializations can lead to more or less optimal weights at the end of training. Therefore, by training the model multiple times starting from different random initializations and then averaging the results, the evaluation of the model's surrogate measure is more reliable. During training, over-fitting was prevented for the first model by using half the data as a validation set and ending training when the prediction error on the validation set increased. A validation set was not used for the second model because of the lack of training data (only one sample per research question). Instead, over-fitting was prevented for the second model by tracking the loss and ending training when the loss began to settle.

4.4 Evaluation of Surrogate Measures

As discussed in Section 2.3, a surrogate measure must meet three criteria (see Section 2.3 for their descriptions). Criteria 1 and 3 can be simultaneously evaluated by looking at the Pearson correlation between the treatment effect on the surrogate measures and the treatment effect on posttest score because a high Pearson correlation between two measures indicates that there is a monotonic linear relationship between them [2], and the linearity implies predictability. The higher the Pearson correlation between treatment effects across all research questions, the more effective the surrogate measure is.

To evaluate Criteria 2, after the surrogate measures were used to determine the treatment effects for the different research questions, a linear regression was fit to predict the treatment effect on posttest given the treatment effect on one of the surrogate measures and an intercept. If the coefficient of the intercept is small and statistically insignificant, then there is no evidence that Criteria 2 was violated. Therefore, the best surrogate measure was determined to be the measure with the highest Pearson correlation between its treatment effects and the posttest treatment effects across all the research questions (Criteria 1 and 3), as long as the measure did not have a significant intercept when its treatment effects were used to predict the posttest treatment effects (Criteria 2).

5. RESULTS

5.1 Evaluation of Surrogate Measures

The treatment effect of each research question was calculated using each surrogate measure described in Sections 4.1 and 4.2. To evaluate whether the surrogate measures met Criteria 1 and 3 from Section 2.3, the treatment effects on each surrogate measure across all the research questions were correlated with the treatment effects on posttest score. Table 3 reports the different surrogate measures, the Pearson correlation [2] of their treatment effects, and the statistical significance of these correlations.

Of all the expert features, correctness and tutoring requested were the only two features whose treatment effects were statistically significantly correlated with the treatment effect on students' posttest scores. Correctness had a positive correlation with posttest score, indicating that students that got the next problem correct on their first try without any support tended to have higher posttest scores than those who did not, and tutoring requested had a negative correlation with posttest score, indicating that students that requested tutoring on the next problem tended to have lower posttest scores than those who did not.

When performing the ablation study to identify the optimal set of expert features for the linear regression used to predict posttest score (Section 4.1, Approach 2), no other feature could be used in combination with correctness to improve the model's predictions. Therefore, using this linear regression to predict posttest was an equivalent surrogate measure to just using correctness as a surrogate measure itself.

When performing the ablation study to identify the optimal set of expert features for the linear regression used to predict treatment effect on posttest (Section 4.1, Approach 3), the highest performing model used tutoring requested and attempt count. Ultimately, this approach was inferior to the other approaches at identifying surrogate measures using expert features.

To evaluate Criteria 2 from Section 2.3, a linear regression was fit for each surrogate measure using data from all the research questions to predict the treatment effect on posttest given the treatment effect on the surrogate measure and an intercept. None of the models had a large or statistically significant intercept. Therefore, the best surrogate measure was simply next-problem correctness.

6. DISCUSSION

Ultimately, next-problem correctness was the best surrogate measure of learning. The treatment effect on next-problem correctness had the highest Pearson correlation with the treatment effect on posttest, and there was no evidence that the treatment effect on next-problem correctness was not zero when the treatment effect on posttest was zero, which satisfies all three criteria discussed in Section 2.3. It was not expected that one of the simplest surrogate measures, which had been used previously despite no empirical evidence to support that choice, would be the best surrogate. One possible reason for why the predictive models did not perform well is that the behavior of students within an experiment could be highly dependent on the material in the assignment. For example, geometry problems might on average take more time to answer than algebra problems, which would make students first response time less informative of their learning because it is in part dependent on the subject matter. Methods like Knowledge Tracing and Performance Factor Analysis, which measure students' mastery of mathematics concepts, take into account the knowledge components of the students' assignments when predicting student performance to compensate for this dependence [3, 13]. By providing the models with more nuanced information about student behavior, it is possible they were picking up on behavioral trends that were not generalizable across experiments. Additionally, the sample size of the data was fairly low. Only 51 research questions were used in this analysis, and it is likely that data from more experiments testing a greater variety of interventions would help the models learn to differentiate between generalizable trends and trends specific to subsets of experiments.

These reasons help to explain what may have caused the models to underperform, but from a different perspective, what caused next-problem correctness to perform so well? It seems likely that next-problem correctness was a strong surrogate because posttest score is simply a different measure of problem correctness. In other words, next-problem correctness is a measure of whether the student got the problem immediately following the intervention correct, and posttest score is a measure of whether the student got a few problems ahead of the intervention correct. It makes sense that two measures that revolve around a student's propensity to answer problems correctly would correlate. This leads to the question: is correctness what matters? If the goal of education is ultimately to give students better, more fulfilling lives, then perhaps test scores are not what a surrogate should measure. There is plenty of evidence of test scores falling short when attempting to correlate them with

Table 3: The Correlations between Surrogate Measure and Posttest Score Treatment Effects

Surrogate Measure	Treatment Effect Correlation with Posttest Score	Correlation p-value
Expert Features as a Surrogate Measure (Section 4.1, Approach 1)		
Correctness	**0.62**	**<0.001**
Tutoring Requested	-0.59	<0.001
No Attempts Taken	-0.01	0.935
Attempt Count	-0.16	0.264
First Response Time	0.04	0.784
Expert Features Used to Predict Posttest Score (Section 4.1, Approach 2)		
Posttest Prediction	0.62	<0.001
Expert Feature Treatment Effects Used to Predict Treatment Effect on Posttest (Section 4.1, Approach 3)		
Treatment Effect Prediction	0.50	<0.001
Deep Learning Posttest Prediction with Mean Squared Error Loss (Section 4.2, Approach 1)		
Posttest Prediction	0.60	<0.001
Deep Learning Posttest Prediction with Treatment Effect Squared Error Loss (Section 4.2, Approach 2)		
Posttest Prediction	0.49	<0.001

things like college and career success. For example, studies have found that SAT scores do not explain any additional variance in college GPA for non-freshman college students after taking into account social/personality and cognitive/learning factors [7].

Perhaps next-problem correctness being the best surrogate measure is an indication that the experiments in ASSISTments are not properly evaluating students' learning. The process of giving students an assignment and then immediately following it with a posttest is likely more a measure of performance rather than learning, which requires long term retention and transfer [23]. The use of posttests immediately following these experimental assignments could be particularly problematic in cases where the assignments themselves require students get three problems correct in a row before completing the assignment. These cases essentially require that students reach similar levels of mastery before evaluating their learning, which likely removes large portions of the effects of the experimental conditions.

6.1 Limitations and Future Work

While in this work next-problem correctness was found the be the best proximal surrogate measure for posttest score, there are some factors that could limit the generalizability of these findings. Firstly, this work uses data entirely from ASSISTments Skill Builder assignments. In these assignments, students are given a series of mathematics problems on the same skill, and are given immediate feedback on each problem as they complete it. next-problem correctness could be especially relevant in this context because the next problem is guaranteed to evaluate the same knowledge components as the previous problem. In assignments where problems require different skills, the problem following an intervention could be only tangentially related to the problem for which the intervention was provided, and thus a student's performance on the next problem would not be a good measure of the effectiveness of the intervention. In the future, using next-problem correctness as a surrogate measure should be evaluated in other kinds of online learning environments, perhaps in contexts where the content students see is chosen adaptively. In this scenario, students will see different problems following an intervention, and combining the next-problem correctness of multiple problems could have positive or negative effects on next-problem correctness's value as a surrogate measure of learning.

Additionally, in this work, only 51 different research questions were used to evaluate the quality of different measures, with a total of 26,060 samples. It is possible that some of the model based attempts at creating a surrogate measure of learning would be more successful if given more data from a wider variety of situations in which A/B testing was performed. Having a larger and more diverse dataset to train the models from also opens up the possibility to train multiple specific models for different subgroups of users or experiments. With the limited data in this work, it was unlikely that splitting the data into subgroups would have helped any of the models. However, with more data it could be the case that a model trained on students with similar backgrounds would be more effective at interpreting behaviors specific to those students. It could also be the case that training a model for a specific type of experiment, for example, experiments that alter the way in which students must answer the question as opposed to experiments that alter the support messages students receive, could improve the model's ability to pick up on different student behaviors associated with these different experiments. In the future, if more data becomes available, models trained on subgroups should be explored.

7. CONCLUSION

In this work, we attempted to derive and validate an effective surrogate measure of learning for use in online learning platforms where rapid A/B testing is used to compare problem-level instructional interventions at scale. To accomplish this, a variety of proximal surrogate measures for posttest score were created through feature engineering, regression, and deep learning. After evaluating each surrogate measure by ensuring it met the criteria for an associative surrogate as described in [10], students' next-problem correctness was determined to be the best surrogate. However, these results could be an indication that the ASSISTments experiments focus on performance rather than learning, and that they should be restructured to measure a more nuanced interpretation of learning.

Follow-up work should be done to validate next-problem correctness as a measure of learning for different types of experiments in different domains and learning environments. Moving forward, using next-problem correctness as a measure of learning within online learning platforms could be an effective way to evaluate students' progress and compare problem-level interventions to each other. We hope this work can help support the educational data mining community by providing methods to create and validate surrogate measures.

8. ACKNOWLEDGEMENTS

We would like to thank Anthony Botelho and Ben Hansen for their thoughtful advice and feedback on the early stages of this work. We would also like to thank NSF (e.g., 2118725, 2118904, 1950683, 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, & 1535428), IES (e.g., R305N210049, R305D210031, R305A170137, R305A170243, R305A180401, & R305A120125), GAANN (e.g., P200A180088 & P200A150306), EIR (U411B190024 & S411B210024), ONR (N00014-18-1-2768), NHI (R44GM146483), and Schmidt Futures. None of the opinions expressed here are that of the funders.

9. REFERENCES

[1] R. S. Baker, N. Nasiar, W. Gong, and C. Porter. The impacts of learning analytics and a/b testing research: a case study in differential scientometrics. *International Journal of STEM Education*, 9(1):1–10, 2022.

[2] J. Benesty, J. Chen, Y. Huang, and I. Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

[3] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[4] C. for Research and J. H. U. Reform in Education. Evidence for essa, 2022.

[5] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

[6] Google. Older unlisted content, 2022.

[7] B. Hannon. Predicting college success: The relative contributions of five social/personality factors, five cognitive/learning factors, and sat scores. *Journal of Education and Training Studies*, 2(4):46, 2014.

[8] N. T. Heffernan and C. L. Heffernan. The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497, 2014.

[9] G. W. Imbens and D. B. Rubin. Rubin causal model. In *Microeconometrics*, pages 229–241. Springer, 2010.

[10] T. Joffe, M. M. Greene. Related causal frameworks for surrogate outcomes. *Biometrics*, 65(2):530–538, 2009.

[11] K. S. Ostrow, D. Selent, Y. Wang, E. G. Van Inwegen, N. T. Heffernan, and J. J. Williams. The assessment of learning infrastructure (ali) the theory, practice, and scalability of automated assessment. In *Proceedings of the sixth international conference on learning analytics & knowledge*, pages 279–288, 2016.

[12] T. Patikorn and N. T. Heffernan. Effectiveness of crowd-sourcing on-demand assistance from teachers in online learning platforms. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*, pages 115–124, 2020.

[13] P. I. Pavlik Jr, H. Cen, and K. R. Koedinger. Performance factors analysis–a new alternative to knowledge tracing. *Online Submission*, 2009.

[14] E. Prihar, A. Haim, A. Sales, and N. Heffernan. Automatic interpretable personalized learning. In *Proceedings of the Ninth ACM Conference on Learning@ Scale*, pages 1–11, 2022.

[15] E. Prihar, A. Moore, and N. Heffernan. Identifying struggling students by comparing online tutor clickstreams. In *International Conference on Artificial Intelligence in Education*, pages 290–295. Springer, 2021.

[16] E. Prihar, T. Patikorn, A. Botelho, A. Sales, and N. Heffernan. Toward personalizing students' education with crowdsourced tutoring. In *Proceedings of the Eighth ACM Conference on Learning@ Scale*, pages 37–45, 2021.

[17] P. R. L. Surrogate endpoints in clinical trials: definition and operational criteria. *Stat Med*, 1989.

[18] J. Renz, D. Hoffmann, T. Staubitz, and C. Meinel. Using a/b testing in mooc environments. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 304–313, 2016.

[19] S. Ritter, A. Murphy, S. E. Fancsali, V. Fitkariwala, N. Patel, and J. D. Lomas. Upgrade: an open source tool to support a/b testing in educational software. In *Proceedings of the First Workshop on Educational A/B Testing at Scale (at Learning@ Scale 2020)*, 2020.

[20] A. Sales, A. Botelho, T. Patikorn, and N. T. Heffernan. Using big data to sharpen design-based inference in a/b tests. In *Proceedings of the Eleventh International Conference on Educational Data Mining*, 2018.

[21] A. C. Sales, E. Prihar, J. Gagnon-Bartsch, A. Gurung, and N. T. Heffernan. More powerful a/b testing using auxiliary data and deep learning. In *International Conference on Artificial Intelligence in Education*, pages 524–527. Springer, 2022.

[22] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[23] N. C. Soderstrom and R. A. Bjork. Learning versus performance: An integrative review. *Perspectives on Psychological Science*, 10(2):176–199, 2015.

Early Prediction of Student Performance in a Health Data Science MOOC

Narjes Rohani
Usher institute,
University of Edinburgh
Narjes.Rohani@ed.ac.uk

Kobi Gal
Ben-Gurion University
University of Edinburgh
kgal@ed.ac.uk

Michael Gallagher
Moray House School
of Education and Sport,
University of Edinburgh
Michael.S.Gallagher@ed.ac.uk

Areti Manataki
School of Computer Science,
University of St Andrews
A.Manataki@st-andrews.ac.uk

ABSTRACT

Massive Open Online Courses (MOOCs) make high-quality learning accessible to students from all over the world. On the other hand, they are known to exhibit low student performance and high dropout rates. Early prediction of student performance in MOOCs can help teachers intervene in time in order to improve learners' future performance. This is particularly important in healthcare courses, given the acute shortages of healthcare staff and the urgent need to train data-literate experts in the healthcare field. In this paper, we analysed a health data science MOOC taken by over 3,000 students. We developed a novel three-step pipeline to predict student performance in the early stages of the course. In the first step, we inferred the transitions between students' low-level actions from their clickstream interactions. In the second step, the transitions were fed into Artificial Neural Network (ANN) that predicted student performance. In the final step, we used two explanation methods to interpret the ANN result. Using this approach, we were able to predict learners' final performance in the course with an AUC ranging from 83% to 91%. We found that students who interacted predominately with lab, project, and discussion materials outperformed students who interacted predominately with lectures and quizzes. We used the DiCE counterfactual method to automatically suggest simple changes to the learning behaviour of low- and moderate-performance students in the course that could potentially improve their performance. Our method can be used by instructors to help identify and support struggling students during the course.

Keywords

Student performance, Neural networks, MOOCs, Explainability, Health data science

N. Rohani, K. Gal, M. Gallagher, and A. Manataki. Early prediction of student performance in a health data science mooc. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 325–333, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115721

1. INTRODUCTION

Today, online learning has greatly changed how people learn. Especially after the Covid-19 pandemic, traditional classrooms are augmented with online activities. In addition, Massive Open Online Courses (MOOCs) have recently made learning more accessible globally to millions of people. Despite the great interest in MOOCs, there are many challenges to their adoption, such as high dropout rates and low learning performance. This is primarily because students need to plan and regulate their learning activities, which can be challenging [20, 24, 11, 23]. Therefore, predicting student performance as early as possible can help teachers provide timely feedback and support to students and inform them of strategies to improve their performance [2].

Although there are many studies on predicting student performance in MOOCs, several important limitations have not yet been addressed [2]. First, most of the previously proposed methods require learner-interaction data of an entire course (from the first to the last day) for prediction. These studies are useful for analysing student performance and behaviour after the course has ended [7, 2]. Conversely, a method with the ability of early prediction of student learning outcomes can help improve student performance [2]. Second, previous work focused only on whether students passed or failed the course [4, 10, 2, 26], while it is also important to identify students with moderate performance. Teachers can potentially help such learners perform better than simply passing the course. Third, most studies on predicting student performance with the use of black-box machine learning models, are difficult to interpret. Therefore, it is hard for teachers to make sense of the predictions and act upon them. As machine learning has been rapidly used in various applications, it has become increasingly important to explain the process that leads to a particular decision [9]. Explanation algorithms can make it easier for teachers to provide personalised feedback to learners. Finally, an important area of education that needs more attention is health data science. According to the National Academy of Medicine, training healthcare professionals who are knowledgeable in both health and data science is highly required, urgent, and challenging [18]. The complexity of teaching interdisciplinary topics to students from diverse backgrounds adds to this difficulty [17]. Therefore, the application of an early student performance predictor to health data science courses can facilitate the much-needed training from which data-literate healthcare experts can emerge.

To address these issues, we propose a three-step pipeline for early prediction of student performance. First, we calculated a transition matrix between different learning actions using a first-order Markov chain representing students' learning processes. Then, the calculated transition matrix was used to classify learners into high- (HP), moderate- (MP), and low-performance (LP) groups using an ANN. Finally, two explanation methods were utilised so as to make the model output more actionable for teachers. The SHAP explanation approach was used to find out which features are important for prediction. Then, we also applied the DiCE method to calculate counterfactual values for LP and MP students, so as to find out how they can improve their learning outcomes.

The proposed pipeline was applied to the Data Science in Stratified Healthcare and Precision Medicine MOOC on Coursera, which includes more than 3000 enrolled students [6]. The results show that students who interacted more with the project, discussion, and lab materials achieved higher final grades. In addition, HP students actively interacted with the video lectures by pausing and replaying the videos. This may indicate that HP students not only watched videos until the end but they also paused, replayed, and sought the video lectures to contemplate the video materials, take notes, or re-watch certain parts of the videos.

The achieved AUC values ranging from 83% to 91% indicate that the method was successful in predicting the performance of health data science students after one week or more of interaction with the course. We also discussed changes suggested by the explanatory method for two students (one LP and one MP) with the help of the course instructor. According to the course instructor, some of the suggested changes are useful for providing personalised feedback to students. The contributions of this study are: i) developing a novel ANN approach for early (after seven calendar days) student performance prediction, ii) employing explanation methods, which may help teachers to provide students with personalised feedback, and iii) applying our approach to an interdisciplinary MOOC in the field of health data science with a high number of enrolled students.

2. RELATED WORK

Prior work for predicting student performance in MOOCs used a variety of different methods. These methods can be classified into tree-based models, linear models, probabilistic models, and Neural Network (NN) approaches. Notable examples include Mbouzao *et al.* [19] who used a tree-based method to predict student success in a MOOC using video interaction data. They analysed data from a McGill University online course on edX over a period of 13 weeks. They defined three metrics based on video interaction data and predicted whether students would pass or fail. The method uses the students' video interaction data after the first and sixth weeks as input. The accuracy of the early prediction was rather low ($\approx 60\%$), while the prediction after the sixth week was more accurate. This result echoes the need to improve the early student performance prediction on MOOCs.

Another example of the application of tree-based methods is the work of Al-Shabandar *et al.* [1]. They analysed behavioural and demographic features of more than 590,000 students from 15 MOOCs in Harvard University's HMedx dataset to predict pass/fail status. They applied several traditional machine learning methods, such as Random Forest (RF), Support Vector Machine (SVM), Naive Bayes (NB), Decision Tree (DT), Logistic Regression (LR), and NN, and showed that RF produces the best performance.

Some papers used linear models to predict student performance. For example, Liang *et al.* [14] applied three linear methods: linear discriminant analysis, LR and Lagrangian SVM (LSVM) to the behavioural data of students in a Data Structures and Algorithms MOOC to predict pass/fail status. They showed LSVM achieved the highest accuracy.

Another group of papers has used probabilistic methods such as NB, Bayes network and Bayesian generalised linear (BGL) models for performance prediction. For example, Cobos *et al.* [5] developed an online tool using various machine learning algorithms such as Boosted LR, RF, NB, NN, SVM and BGL to predict pass/fail status. The tool works based on analysing behavioural and video interaction data of students collected from 15 different MOOCs in social science and science fields. It was found that BGL is the best model as it can be trained quickly and gives stable results (AUC between 60% and 80%).

Numerous works have used NN such as MultiLayer Perceptron (MLP) or ANN, Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) to predict student performance. For example, Kőroesi and Farkas [13] developed an RNN model based on raw clickstream data that is suitable for both regression and multiclass classification of weekly student performance. They used the Stanford Lagunita dataset, which consists of log data from 130,000 students, and took the final quiz score as the output for the regression problem, while the students were divided into 10 levels based on their final scores for the classification problem. Although their model does not require a feature engineering step, the classification accuracy of the best model (using features of all weeks) is low (around 55%).

Qu *et al* [22] analysed a C programming MOOC with 1525 learners. They focused only on the log data of the programming tasks. Features such as submission times and order of submissions were used to predict student performance using an MLP. The results show that failing students have an obvious sequence pattern when trying to solve programming tasks, while the behaviour of passing students is less straightforward. The authors also developed an MLP with LSTM and discriminative sequential pattern mining to capture learners' behavioural patterns and predict their performance. These NN-based models are black-box and would require an additional explanatory step to help teachers understand the results of these models [25].

The current approaches do not provide satisfactory performance in early student performance prediction [2]. Also, teachers need to identify LP students as soon as possible

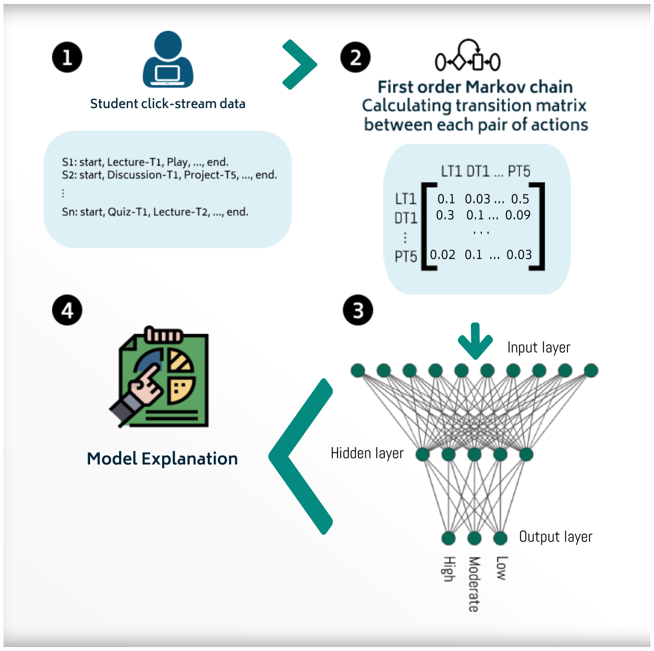

1
Student click-stream data
2
First order Markov chain
Calculating transition matrix
between each pair of actions
S1: start, Lecture-T1, Play, ..., end.
S2: start, Discussion-T1, Project-T5, ..., end.
⋮
Sn: start, Quiz-T1, Lecture-T2, ..., end.
LT1 DT1 ... PT5
LT1 0.1 0.03 ... 0.5
DT1 0.3 0.1 ... 0.09
PT5 0.02 0.1 ... 0.03
4
Model Explanation
3
Input layer
Hidden layer
Output layer
High
Moderate
Low

Figure 1: Schema of our approach

to help them adopt effective learning strategies. Moreover, none of the previous NN-based studies used explanatory methods to make the result more actionable and interpretable for teachers. The interpretation of the prediction is key for teachers since they need to understand the learning behaviour of students to write personalised feedback.

3. METHODOLOGY

Figure 1 shows the schema of our approach for early student performance prediction. First, for each learner, a sequence of learning actions during a set time window (e.g. one calendar week) was extracted. These represent the lowest level actions carried out by the student (e.g., playing a video lecture, submitting an assignment, and so on). The set of possible learning actions can be defined based on clickstream data and the course design (See Section 4). Second, the transition probability between each pair of actions was computed using a first-order Markov chain. Third, an ANN was trained to predict students' performance levels (LP, MP, and HP) given a student's transition matrix. Finally, the DiCE and SHAP methods were employed to explain the model decision in order to help teachers write personalised feedback.

Let $A = \{a_0,\ a_1,\ a_2,\ ...,\ a_N, a_{N+1}\}$ denote Markov states, where $a_0 = start$, $a_{N+1} = end$, each a_i is a learning action, and N is the number of all learning actions. Assume $S_k = (s_k^{(0)},\ s_k^{(1)}, ...,\ s_k^{(n)})$ be the sequence of actions for kth student in a time frame, in which $s_k^{(0)} = start$, $s_k^{(n)} = end$ and $s_k^{(t)} \in A$ be the action that the kth student has done in the tth time of the sequence of actions. The sequence of actions can be seen as a trajectory between states in the Markov chain and is used to estimate the transition probability between Markov states. Based on the Markov chain, the transition probability matrix for the learning process for kth student is $P_k = [p_k(i,\ j)]_{i,j\in\{0,...,N+1\}}$ calculated by Formula (1).

$$p_k(i,\ j) = \frac{|s_k^{(t)} = a_i\ and\ s_k^{(t+1)} = a_j|}{\sum_{l\neq i} |s_k^{(t)} = a_i\ and\ s_k^{(t+1)} = a_l|}, \qquad (1)$$

where $|.|$ is the count function. In Formula (1), the number of transitions from a_i to a_j is divided by the number of all transitions emanating from a_i. The transition probability matrix is calculated for each student separately. Although the action sequences for each student can change, the transition probability matrix for all students has the same dimension of $(N + 2) \times (N + 2)$. Note that if kth student never commits the transition from action a_i to action a_j, $p_k(i, j) = 0$.

For each student, the transition probability matrix of all actions in the time frame served as input to the ANN. We employed ANN as the state of the art in predicting student performance; they have been shown to outperform traditional methods [3]. The ANN model includes input, hidden, and output layers. The hidden layer computes the latent features extracted from the input layer using $ReLU(x)$ as the activation function. The output layer has three neurons to compute the probability of the input belonging to each of the three classes (LP, MP, and HP) using $Sigmoid(x)$ as the activation function. The final grade can be mapped to LP, MP, or HP categories, or more finer-grained categories, based on instructors' preferences. A set of hyperparameters were used for finding the best ANN architecture. The values tested for the number of hidden neurons were 5, 10, 15, 20, 50, 100, and 200. The batch size values tested were 4, 8, 16, 32, 64, and 128. The number of epochs tested was 5, 10, 15, and 20. To train the model, the categorical cross-entropy loss (CCE) was computed on each batch of data and the weight values were updated based on ADAM optimizer [12] after feeding each batch. The model performance was evaluated on the test data using the Area Under ROC Curve (AUC). To evaluate the performance of the predictive model objectively, we used 5-fold stratified Cross-Validation (CV) and 20% of the training data were considered as validation data (changed in each fold) for tuning the hyperparameters.

3.1 Interpretability

We used the SHAP (SHapley Additive exPlanations) method [16] to select the most important features in predicting student performance. In the SHAP method, Shapley values are calculated for each transition probability (features) and the transitions with the highest Shapley values were considered the most important features that contribute the most to the model prediction. The Shapley value for a transition from action a_i to action a_j is $\phi_{a_i\rightarrow a_j}$ and is defined in Formula (2). Based on formula (2), $\phi_{a_i\rightarrow a_j}$ is the average improvement of the model by adding this feature (transition from action a_i to a_j) to all models considering different possible features. Herein, a feature is a transition from one action to another.

$$\phi_{a_i\rightarrow a_j} = \sum_{S\subset M-\{i\}} \frac{|S|!(|M|-|S|-1)!}{M!}(f(S\cup i) - f(S)), \qquad (2)$$

where M is the set of all features and $f(S)$ is the performance of model based on subset S of features. Since the features are the set of all possible transitions between Markov states, $|M| = (N+2) \times (N+2)$. After calculating $\phi_{a_i \to a_j}$ for all $a_i,\ a_j \in A$, we ranked them based on their importance and selected the most important features.

In order to make the model more actionable for teachers, we used the DiCE [21] method for calculating counterfactual examples (CFs) to explain the conditions that can potentially change the students' performance. Each CF is a set of changes (increase or decrease) in some transitions between learning actions. An example of a set of CFs is increasing the transition from Lecture A to Quiz B and decreasing the transition from Video pause to Video end. A good set of CFs should be efficient, proximal, and diverse. The efficiency of CF means that applying those changes in the students' learning process may lead to higher performance. Proximity means that the suggested changes should be close to the current learning process of the student; i.e. the CFs suggesting huge changes in students' current learning process are not practical. Finally, the diversity of the CFs denotes that the set of proposed changes in CFs should have the highest variety, so that the student can have different options.

Consider an LP student with a transition matrix of P. A reasonable CF can be the transition matrix P' that has the same dimension and values as P, but with subtle changes in some of the elements. Assume this student has a high transition probability from Video end to Quiz A, but a lower transition from Video end to Quiz B. Suppose that HP students proceed to Quiz B after the VideoEnd action with a high probability. In this case, recommending this student visit Quiz B after the VideoEnd action may increase the performance of the student. To this aim, for each LP or MP student with a transition matrix of P, the set of $P'_1, P'_2, ..., P'_m$ counterfactual transition matrices are selected such that the following loss function is minimised.

$$CF(P) = argmin_{P'_1, P'_2, ..., P'_m} \sum_{i=1}^{m} L(f(P'_i), y^*) \tag{3}$$

$$+ \frac{\lambda_1}{m} \sum_{i=1}^{m} dist(P'_i, P) - \lambda_2 dppDiversity(P'_1, P'_2, ..., P'_m)$$

In Formula (3), $f(P'_i)$ is the predicted performance of the student considering P'_i as his/her transition matrix, y^* is the ideal performance, L is the distance between prediction for P'_i and the ideal performance. $dist$ is the Manhattan distance of two transition matrices, $dppDiversity$ is the diversity of counterfactual transition matrices which is defined based on Formula (4), and λ_1, λ_2 are the regularization terms to balance three terms of loss functions.

$$dppDiversity(P'_1, P'_2, ..., P'_m) = det(K) \tag{4}$$

$$k(i, j) = \frac{1}{dist(P'_i, P'_j)} \tag{5}$$

where i, j is any two CFs, and $det(K)$ is the determinant of the matrix K which its elements are defined based on Formula (5). Consequently, three terms in calculating $CF(P)$ represent the constraints for selecting good CFs. To be specific, minimising $\sum_{i=1}^{m} L(f(P'_i), y^*)$ guarantees the efficiency of CF to be chosen in a way that may lead to high performance. Also, minimising $\frac{\lambda_1}{m} \sum_{i=1}^{m} dist(P'_i, P)$ narrows down the CFs to the set of transition probabilities that are close to the current learning process. Finally, $dppDiversity(P'_1, P'_2, ..., P'_m)$ ensures the diversity of CFs. For example, for each LP or MP student with a transition matrix of P, various random proximal transition matrices P' with some changes in some of the elements are considered. Among different possible CFs, the set of m transition matrices which is highly probable in high-performance students and leads to the minimum $CF(P)$ are selected. The selected CFs such as an increase or decrease in some transition values, can potentially be used to guide students towards improving their performance.

4. APPLICATION TO HEALTH DATA SCIENCE MOOC

We applied our approach to data from the Data Science in Stratified Healthcare and Precision Medicine (DSM) MOOC on Coursera, for the period between April 2018 and April 2022 [6]. Over this period, 3,527 learners were enrolled (38% male, 28% female, and 34% unknown) with at least one learning action. The course completion rate for these students is 38%. DSM is a self-paced 5-topic MOOC with a total of 43 videos, 13 reading materials, five quizzes, one programming assignment and one peer-review/project assignment. The course assessment includes a quiz for each topic, as well as a programming assignment for the third topic and a peer-reviewed assignment for the last topic. The final grades were calculated (out of 100) by the weighted average of all quiz and assignment scores (each quiz weight = 10%, programming assignment weight = 20%, and peer-reviewed assignment weight = 30%). Upon the course instructor's request, we grouped students into three performance groups. An LP group (final grade < 50; i.e. student failed the course), which included 62% of students; an MP group (50 $\leq$ final grade < 80), which included 21% of students; and an HP group (final grade $\geq$ 80), which included 16% of students.

We used anonymised data and have received institutional ethics approval for this research. All 3,527 enrolled learners with at least one action were used for the analysis. The considered actions include starting to watch a video lecture (`Video-Start`), playing a video lecture (`Play`), watching a video lecture until the end (`Video-End`), skipping forward or backwards throughout a video lecture (`Seek`), pausing a video lecture (`Pause`), changing the volume of a video lecture (`Volume-Change`), changing the subtitle of a video lecture (`Subtitle-Change`), downloading a video lecture (`Download-Video`), downloading video lecture subtitle (`Download-Subtitle`), changing the play rate of a video lecture (`Playback-Rate-Change`), visiting the main page of the video lecture i (`Lecture-Topic`$_i$), engaging with discussion forum i or posting a question on the forum (`Discussion-Topic`$_i$), engaging with general discussion forums (`Discussion-General`), engaging with reading material i (`Reading-Topic`$_i$), engaging with quiz i such as visiting the quiz page or submitting the quiz (`Quiz-Topic`$_i$), engaging with lab materials of topic i (`Lab-Topic`$_i$), and engaging with the peer-reviewed assignment such as visiting the project or project submission (`Project-Topic`$_i$). The i is a topic number ranging from 1 to 5.

For all performance groups, we computed the percentage of students in each group that carried out each learning action, denoted RAP (Relative Action Presence). Interestingly, more than 80% of the LP students interacted with the first topics more than the topics towards the end of the course. The RAP score for LP students' learning actions decreases as the course advances. On the other hand, almost all HP students were involved in assessment-related actions, such as projects, lab work and quizzes. MP students have similar RAP scores to HP students, although the order of frequent actions is slightly changed.

For example, the project topic5, which accounts for 30% of the total score, has the highest RAP (almost 100%) for the HP students, a low RAP (about 10%) for LP and a relatively high RAP (about 80%) in the MP students. Also, RAP for Discussion-General is relatively high (about 75%) in the HP, medium (about 55%) in the MP and low (0.25%) in the LP group. In general, the majority (about 80%) of HP students were involved in two-thirds of the activities, while the majority (about 80%) of the LP students were involved in one-third of the actions, highlighting the fact that the HP students were involved in more actions than the LP. Furthermore, since the RAP of lab work, projects, and quizzes are much larger among HP students compared to the LP group, it can be concluded that HP students focused on assessment-related actions. It should be noted that the low RAP of action in a group could be caused by a high dropout of students (fewer students continued the course) or low engagement of students that continued the course.

Even for the actions of the first topic, the RAP of HP is greater than MP students, and the RAP of the MP is greater than the LP group. These differences show that students' performance can be predicted based on their level of interaction with the first topic. As the course progresses, the difference in RAP increases between the HP and the MP, and between the MP and the LP; i.e. the differences between the groups become more pronounced as the course progresses.

4.1 Learning Processes

To shed more light on the differences in the learning process between the HP, MP, and LP groups, the transition probability matrices for each of the groups of students were calculated using First-order Markov models [8]. The difference between the transition probability matrices of each pair of groups is shown in Figure 2.

One difference between the students of the HP and LP groups is how students interacted with videos. The red colours in the pause and seek columns show that the HP students are more inclined to use the pause and seek actions than the LP ones (Figure 2 a). Consequently, seeking and pausing videos, which may involve contemplating the video material, making notes, or re-watching certain parts of the lecture, is a helpful action that may lead to better performance. Conversely, it can be concluded that finishing a video on its own is not an indicator of a good comprehension of the concepts presented in the video.

Another difference between the HP and LP groups is how students transitioned from the video-download action. After doing this action, students in the HP group proceeded mainly to the main page of lecture topics 5 and 4, while students in the LP group proceeded to the main page of lecture topics 1 and 2 (Figure 2 a). A similar trend appears when comparing the matrices of the MP and LP groups (Figure 2 b). The transitions from VideoEnd to lecture topics show that HP students are more likely to go to lecture topics 5 and 4, while LP ones prefer to move to lecture topic 1 (Figure 2 a). LP students engage more with actions in the first topics, while HP and MP students focus more on the last topics, which contribute more to the overall score.

Interestingly, after visiting the general discussion forum, HP students mostly move to discussion topic 5, while the LP students mainly moved to discussion topic 1 (Figure 2 a). Also, the high probability of transition from discussion topic 5 to itself, and project topic 5 to itself, for the HP and MP students when compared to the LP students, support that the HP and MP were engaged with and discussed project topic 5 more than the LP students.

There are a few differences between the HP and MP groups. The most obvious difference is the higher likelihood of using seek and pause actions among the HP compared to the MP students (Figure 2 b), which supports the hypothesis that seek and pause can lead not only to an acceptable but also to a high final grade. Another difference is that the HP students are more likely to select discussion topic 5 after going to the discussion area, while students from the MP group are more likely to stay in the general discussion forum, which includes discussion related to the course but not strictly related to a particular weekly topic (Figure 2 b).

4.2 Early Prediction of Student Performance

In this study, we set week as the time window; therefore, the model is able to predict students' final performance after seven calendar days or more. Five prediction models were built and trained using clickstream data available up to each calendar week. The best values of the hyperparameters were determined based on the performance of the models on the validation data. Accordingly, the number of hidden neurons, epoch size, and batch size were set to 200, 10, and 128, respectively. Mean AUC values were averaged over 10 replications of the 5-fold CV.

Table 1: AUC of the model to predict HP, MP and LP after each calendar week (7 days).

	AUC				
Time window	LP	MP	HP	Micro	Macro
First week	0.78	0.65	0.74	0.83	0.72
First two weeks	0.89	0.76	0.84	0.89	0.83
First three weeks	0.91	0.70	0.83	0.87	0.81
First four weeks	0.93	0.74	0.87	0.90	0.85
First five weeks	0.94	0.80	0.88	0.91	0.87

Table 1 shows the AUC of each model for predicting students performance. It is obvious that the AUC increases over time for the prediction of the LP and HP students. Based on Table 1, the AUC for the prediction of the MP students decreases slightly in the first three weeks' analysis in comparison with the first two weeks' analysis. This could be due to the different behaviour of MP students in the third week compared to their behaviour in the other weeks. A possible explanation for this difference may be the pro-

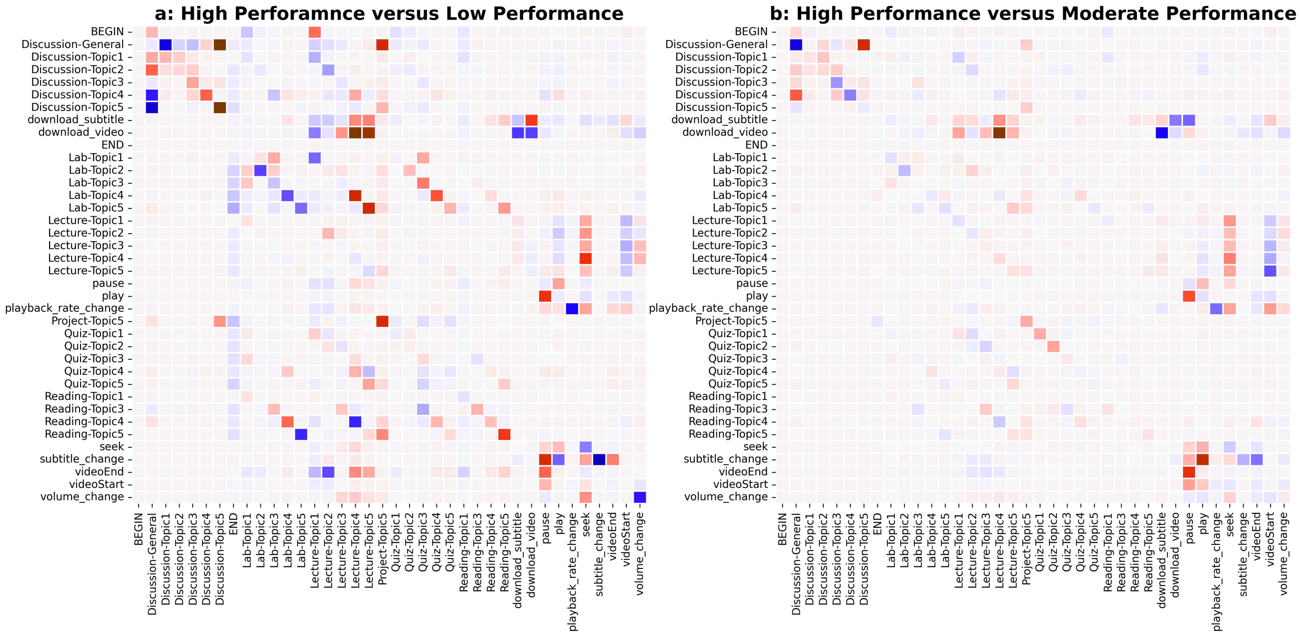


Figure 2: The y-axis and x-axis represent the source and destination of the transitions, respectively. The values range from -1 (blue) to 1 (red), centred at zero (grey), and the intensity of colour shows the magnitude of the difference. The red elements in (a) represent that the probability of a transition between a pair of actions is higher in HP students than LP students, while blue elements show a lower probability of the transition in HP than in LP students. (b) Red cells indicate a higher probability of a transition in HP than MP students, while blue elements are the reverse.

gramming assignment as an assessment for the third week, which might have an impact on the MP students' behaviour. This decrease in the AUC of the prediction of MP based on the first three weeks affects the overall AUC value.

Although the model based on the first five weeks achieved excellent AUC value (91%), indicating its great potential in stratifying students, the model based on the first calendar week also succeeded to classify students with a good AUC (83%), showing that students' performance can be predicted with good accuracy from their actions in the first seven days (See Table 1). Moreover, the performance of each model in predicting the LP students is better than that of the MP or HP students. This could be due to the larger group size, and thus more training data from the LP students, or the better discrimination of the definition of the LP students (score from 0 to 50) compared to the two other groups. We used the zeroR model as a baseline similar to the related work [27, 15]. The proposed method significantly outperforms the zeroR model baseline (AUC = 0.5, accuracy = 0.62).

4.3 Explanation and Important Features

The SHAP method was applied to estimate the importance of features based on their influence on the predictive model of the first week. The most important feature is the transition from video pause to play, which has a large, medium and small impact on the prediction of the HP, MP, and LP students, respectively. The top 10 important features include transitions between play, pause, seek, videoStart, and videoEnd, indicating the high impact that interaction with videos has on their performance. Both transitions from pause to play and from play to pause are highly important, with a relatively even impact of *play* → *pause* in the prediction of each group and a greater impact of *pause* → *play* in predicting the HP students, highlighting that even if all students paused videos at the same rate, HP students resumed videos much more frequently than others. The same is true for the transitions *seek* → *pause* and *pause* → *seek*; thus, resuming videos after a pause or seek is a better indicator of the HP students than pausing or seeking itself. To assess the values of the most important features in each group, their relative occurrence was calculated for each group of students. All the top features have high, medium and low relative occurrence among HP, MP, and LP students, respectively. Although there are many more students in the LP group and only a few students in the HP group, the relative frequency of the HP students is much higher for the top features, which shows that the total number of actions (transitions) for this small number of HP students was greater than the total number of actions for the large population of LP students. Consequently, this can be considered as an indicator of the diligence of the HP students, the mediocre effort of the MP students, and the minimum number of actions of the LP students.

We also tried to select the top 300 important features using the SHAP method to train the models. This resulted in micro-average AUC values of 84%, 88%, 88%, 91%, and 91% for the predictive models in weeks 1 to 5. However, the improvement in model performance was insignificant, which suggests that our method was able to extract important information from the sparse input features.

In the final analysis, we employed the DiCE method to sug-

Table 2: Suggested changes that can lead to increasing the performance of students with low and moderate performance. As an example, increasing $Lab - Topic1 \rightarrow Quiz - Topic5$ means increasing the transition from lab material topic 1 to quiz topic 5.

Student	Suggested changes	
Student1 (Current group: LP)	Increase $Lab - Topic1 \rightarrow Quiz - Topic5$	$Lecture - Topic1 \rightarrow Videoseek$
	$videoStart \rightarrow Discussion - Topic5$	$Videopause \rightarrow Discussion - Topic3$
	$Lab - Topic3 \rightarrow Project - Topic5$	$Lecture - Topic4 \rightarrow Lecture - Topic1$
	$Lecture - Topic5 \rightarrow Lab - Topic5$	$Quiz - Topic5 \rightarrow Videoplay$
	$Quiz - Topic5 \rightarrow Lecture - Topic3$	
Student2 (Current group: MP)	Increase $Lecture - T3 \rightarrow Lab - Topic5$	$Discussion - Topic3 \rightarrow Discussion - Topic5$

gest potential changes for LP and MP students that could improve their performance. Table 2 shows example results of the method for two students, one in the LP group and one in the MP group. Below are some interpretations based on the suggested changes in addition to the course instructor's discussion around the suggested changes.

Student 1 (an LP student): It seems that this student had more trouble with the theoretical questions than with the programming questions in the quizzes. Therefore, they should watch video lectures and take notes before taking the quizzes. Also, this student is advised to focus more on the programming lab in Topic 1, before taking Quiz 5. According to the course instructor, this is a meaningful recommendation, as this lab can support refreshing fundamental programming knowledge, which aids in answering programming questions. Another recommendation that is meaningful according to the course instructor is around using discussion forums more. In particular, the algorithm highlights engaging with the discussion forums for Topics 3 and 5 upon watching lecture videos. In online education, posting questions in the forums and reading existing discussions is a good strategy for clarifying questions that may arise when watching videos. Some suggestions, however, are harder to decipher, according to the course instructor. In particular, it is unclear why it is recommended to engage with the programming lab in Topic 3 before attempting the peer-reviewed assessment, given that they cover very different concepts.

Student 2 (an MP student): By increasing only two transitions, he/she can become an HP student. It can be deduced that the student needs to work more on the topic of lecture 3 and then on topic 5. This student can improve his/her performance if he/she spends more time on lab material and discussions 3 and 5. According to the course instructor, it is not a surprise that topics 3 and 5 are highlighted here, as these two topics are strongly related to the programming and the peer-reviewed assignment. The recommendation, however, to increase the transition from the lecture in topic 3 to the programming lab in topic 5 is somewhat unexpected, as the two topics cover rather different content. The instructor has speculated that students might benefit from refreshing knowledge related to network analysis in topic 3 when learning new concepts around graph data in topic 5, even though this link is not made evident in the course design. This is an interesting hypothesis to investigate in the future.

5. DISCUSSION

We proposed a novel approach for early predicting student performance based on their learning process. Our method, a combination of ANN and Markov chain, classified learners into three performance groups with AUC ranging from 83-91%. The results showed that even after only one week of interaction with the course, our method can predict final performance with reasonable accuracy. We also used SHAP and DiCE explanation methods to identify important features and suggest changes for LP and MP students to potentially improve their performance. The proposed pipeline can be used for different courses towards providing early and personalised interventions to students. Since artificial intelligence methods are not error-free, they are only an assistant for teachers to provide them with processed information. Ultimately, it is teachers who write personalised feedback for students by analysing the method results.

Learner behaviour in the health data science MOOC shows that interacting with video lectures, such as pausing or replaying a video, which may be related to contemplating on the material, taking notes, or re-watching certain parts result in a higher final grade. Investing more time in learning materials related to key assessments (i.e. lab materials and content from topics 3 and 5) also leads to higher grades. Our analysis indicates that LP students lose motivation after attending Topic 3, while their engagement with Topic 1 materials is high. Our recommendation is to divide large assessments into small tasks that a student can work on each week, so as to motivate them and improve their performance.

A limitation of this work is that enrolled learners in MOOCs have different motivations; therefore, the definition of performance criteria is conceptually controversial. We focused on the final grade as an indicator of learning performance. Further research is needed to define a new performance criterion that considers learner motivation as well as the final grade. Since the method does not depend on course design and can be used for MOOCs with a different number of topics and learning materials, it needs to be applied to multiple courses with different designs, contexts, and sample sizes to assess its generalisability. The explanation step of the pipeline can be improved with textual and visual explanations based on educational learning theories. We have shown that some of the suggestions by explanation methods make sense for instructors but there are several recommendations which are not clear enough. Further studies are needed to process the output of the explanation step for making them more consistent with learning theories and teachers' prior knowledge about the course.

6. ACKNOWLEDGMENTS

This work was supported by the Medical Research Council [grant number MR/N013166/1].

7. REFERENCES

[1] R. Al-Shabandar, A. Hussain, A. Laws, R. Keight, J. Lunn, and N. Radi. Machine learning approaches to predict learning outcomes in massive open online courses. In *2017 International joint conference on neural networks (IJCNN)*, pages 713–720. IEEE, 2017.

[2] A. Alhothali, M. Albsisi, H. Assalahi, and T. Aldosemani. Predicting student outcomes in online courses using machine learning techniques: A review. *Sustainability*, 14(10), 2022.

[3] Y. Baashar, G. Alkawsi, A. Mustafa, A. A. Alkahtani, Y. A. Alsariera, A. Q. Ali, W. Hashim, and S. K. Tiong. Toward predicting student's academic performance using artificial neural networks (anns). *Applied Sciences*, 12(3):1289, 2022.

[4] Y.-C. Chiu, H.-J. Hsu, J. Wu, and D.-L. Yang. Predicting student performance in moocs using learning activity data. *J. Inf. Sci. Eng.*, 34(5):1223–1235, 2018.

[5] R. Cobos and L. Olmos. A learning analytics tool for predictive modeling of dropout and certificate acquisition on moocs for professional learning. In *2018 IEEE international conference on industrial engineering and engineering management (IEEM)*, pages 1533–1537. IEEE, 2018.

[6] Coursera. Data science in stratified healthcare and precision medicine | coursera. https://www.coursera.org/learn/datascimed. Accessed: Dec. 5, 2022.

[7] J. E. M. Fotso, B. Batchakui, R. Nkambou, and G. Okereke. Algorithms for the development of deep learning models for classification and prediction of behaviour in moocs. In *2020 IEEE Learning With MOOCS (LWMOOCS)*, pages 180–184. IEEE, 2020.

[8] R. Gatta, J. Lenkowicz, M. Vallati, E. Rojas, A. Damiani, L. Sacchi, B. De Bari, A. Dagliati, C. Fernandez-Llatas, M. Montesi, A. Marchetti, M. Castellano, and V. Valentini. pminer: An innovative r library for performing process mining in medicine. In A. ten Teije, C. Popow, J. H. Holmes, and L. Sacchi, editors, *Artificial Intelligence in Medicine*, pages 351–355, Cham, 2017. Springer International Publishing.

[9] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang. Xai—explainable artificial intelligence. *Science robotics*, 4(37):eaay7120, 2019.

[10] N. I. Jha, I. Ghergulescu, and A.-N. Moldovan. Oulad mooc dropout and result prediction using ensemble, deep learning and regression techniques. In *CSEDU (2)*, pages 154–164, 2019.

[11] K.-J. Kim and C. J. Bonk. The future of online teaching and learning in higher education. *Educause quarterly*, 29(4):22–30, 2006.

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] G. Kőrösi and R. Farkas. Mooc performance prediction by deep learning from raw clickstream data. In *International Conference on Advances in Computing and Data Sciences*, pages 474–485. Springer, 2020.

[14] K. Liang, Y. Zhang, Y. He, Y. Zhou, W. Tan, and X. Li. Online behavior analysis-based student profile for intelligent e-learning. *Journal of Electrical and Computer Engineering*, 2017, 2017.

[15] D. Litman and K. Forbes-Riley. Predicting student emotions in computer-human tutoring dialogues. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 351–358, 2004.

[16] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[17] A. S. Lundervold and A. Lundervold. An overview of deep learning in medical imaging focusing on mri. *Zeitschrift für Medizinische Physik*, 29(2):102–127, 2019.

[18] M. E. Matheny, D. Whicher, and S. T. Israni. Artificial intelligence in health care: a report from the national academy of medicine. *Jama*, 323(6):509–510, 2020.

[19] B. Mbouzao, M. C. Desmarais, and I. Shrier. Early prediction of success in mooc from video interaction features. In *International Conference on Artificial Intelligence in Education*, pages 191–196. Springer, 2020.

[20] P. M. Moreno-Marcos, C. Alario-Hoyos, P. J. Muñoz-Merino, and C. D. Kloos. Prediction in moocs: A review and future research directions. *IEEE transactions on Learning Technologies*, 12(3):384–401, 2018.

[21] R. K. Mothilal, A. Sharma, and C. Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 607–617, 2020.

[22] S. Qu, K. Li, B. Wu, S. Zhang, and Y. Wang. Predicting student achievement based on temporal learning behavior in moocs. *Applied Sciences*, 9(24):5539, 2019.

[23] N. Rohani, K. Gal, M. Gallagher, and A. Manataki. Discovering students' learning strategies in a visual programming mooc through process mining techniques. In *Process Mining Workshops: ICPM 2022 International Workshops, Bozen-Bolzano, Italy, October 23–28, 2022, Revised Selected Papers*, pages 539–551. Springer, 2023.

[24] S. L. Schneider and M. L. Council. Distance learning in the era of covid-19. *Archives of dermatological research*, 313(5):389–390, 2021.

[25] V. Swamy, B. Radmehr, N. Krco, M. Marras, and T. KÃ¤ser. Evaluating the explainers: Black-box explainable machine learning for student success prediction in MOOCs. In A. Mitrovic and N. Bosch, editors, *Proceedings of the 15th International Conference on Educational Data Mining*, pages 98–109, Durham, United Kingdom, July 2022. International Educational Data Mining Society.

[26] B. Xiao, M. Liang, and J. Ma. The application of cart algorithm in analyzing relationship of mooc learning behavior and grades. In *2018 International Conference on Sensor Networks and Signal Processing (SNSP)*, pages 250–254. IEEE, 2018.

[27] A. Zohair and L. Mahmoud. Prediction of student's performance by modelling small dataset size. *International Journal of Educational Technology in*

Higher Education, 16(1):1–18, 2019.

Self-Assessment Task Processing Behavior of Students in Higher Education

Regina Kasakowskij
Research Institute CATALPA
FernUniversität in Hagen,
Germany
regina.kasakowskij@fernuni-hagen.de

Joerg M. Haake
Research Institute CATALPA,
FernUniversität in Hagen,
Germany
joerg.haake@fernuni-hagen.de

Niels Seidel
Research Institute CATALPA
FernUniversität in Hagen,
Germany
niels.seidel@fernuni-hagen.de

ABSTRACT

Improving competence requires practicing, e.g. by solving tasks. The Self-Assessment task type is a new form of scalable online task providing immediate feedback, sample solution and iterative improvement within the newly developed SAFRAN plugin. Effective learning not only requires suitable tasks but also their meaningful usage within the student's learning process. So far, learning processes of students working on such Self-Assessment tasks have not been studied. Thus, SAFRAN was extended with activity logging allowing process mining. SAFRAN was used in a first-year computer science university course. Students' behavior was clustered and analyzed using log data. 3 task completion behavior patterns were identified indicating positive, neutral or negative impact on task processing. Differences in the use of feedback and sample solutions were also identified. The results are particularly relevant for instructors who can tailor adaptive feedback content better to its target group. The analytics approach described may be useful for researchers who want to implement and study adaptive and personalized task processing support.

Keywords

Sequence pattern analysis, Self-Assessment tasks, students task processing behavior, distance learning.

1. INTRODUCTION

Teaching can be described as a sequence of teaching-learning processes planned and designed by teachers [31]. As a central instrument for planning, controlling, and evaluating these processes, exercises in the form of tasks have long played a significant role in the learning context [24]. They serve to promote learning effectiveness by helping to apply and consolidate knowledge learned. This applies to both traditional and multimedia learning opportunities. Online tasks in particular offer many advantages. Students are able to work on them independent of location and time and receive immediate feedback on their performance. Students are able to learn and test their knowledge independently and, in some cases, self-directed, without the direct instruction and support of teachers as well as other students. However, there are also limitations. Learning in a virtual environment is, for example, apart from live sessions with teachers, predominantly an asynchronous learning process. When working on tasks students are left to their own devices and must show initiative if they do not understand something. This is a hurdle that not every student can overcome, which often leads to incorrect understanding or even abandonment of the task [16, 7]. Students can often receive feedback after completing a task, but this may not be helpful for or used by every student [15].

1.1 Self-Assessments as a competency-enhancing task type

The use of competency-enhancing (complex and problem-oriented) tasks that students can complete independently is intensively discussed by researchers and teachers [18, 22]. But not all traditional assessment strategies can be applied to online courses. For example, there are differences in the way a task is presented, the type of task, the complexity of the task, and appropriate support during the task. However, most tasks are at the lower two levels of Bloom's taxonomy [4] and are thus of lower complexity.

Recent developments enable scalable competency-enhancing tasks in online environments [12, 30, 8, 27, 29]. Self-Assessments can be used to set complex tasks; thus, they belong to the competency-enhancing task types. They are a special type of tasks with which students are able to evaluate their own solution based on assessment criteria and thus assess their own performance without third parties having to act as mediators. Students shall become a feedback provider themselves and gain an understanding of what a good work in the subject looks like, assuming they can accurately evaluate their own solution [2, 5]. But Self-Assessment tasks alone are not self-explanatory in case of an error in one's own solution. They are difficult to scale up in a virtual learning environment. Therefore, additional feedback is needed, which helps students to correct their own solution. This problem was identified and solved by [12] and implemented in a Moodle virtual learning environment [30]. In this approach, students begin the Self-Assessment process by selecting a relevant learning task to complete. Then the task, including instructions, is displayed and students are asked to create and submit a solution. After that, a list of assessment criteria set by the instructor is presented, a sample solution is provided on demand, and students are asked to evaluate the submitted solution. After the students have evaluated the solution using the provided assessment criteria, feedback based on their Self-Assessment is automatically

R. Kasakowskij, J. M. Haake, and N. Seidel. Self-assessment task processing behavior of students in higher education. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 334–341, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115707

selected from a feedback database defined by the trainer and presented to the students. Using the feedback, students can then reflect on the quality of their learning products and improve their solution in a new iteration (create, upload, self-assess the improved solution again, receive feedback, and accept or reject another iteration) until they self-assess their solution as correct or good enough or decide to complete the exercise [12, 13].

This type of online task is well suited to examine the task processing behavior patterns of students and their handling of feedback and sample solutions because, on the one hand, it consists of a reasonable set of possible task processing steps. On the other hand, it can be used to set and solve complex tasks of varying difficulty. For this reason, the process was adopted for use in a LMS.

1.2 Self-Assessment plugin: Improvement and Implementation

The prototypical implementation from [30] is a Moodle quiz type plugin. Since this form of quizzes was somewhat cumbersome in the implementation of the intended iterative process of working on Self-Assessment tasks and slowed it down, the support of the process was re-implemented as a Moodle activity plugin named SAFRAN (Self-Assessment with Feedback RecommendAtioNs), and thereby a simpler and faster editing process enabled. In addition, students were able to write their solution directly into an editor field, which was previously only possible by uploading .pdf, .png and .jpg files. Furthermore, additional information, such as process steps, clicks on feedback links, clicks on sample solutions and ratings of feedbacks with additional reasons for negative feedback, were saved in a log.

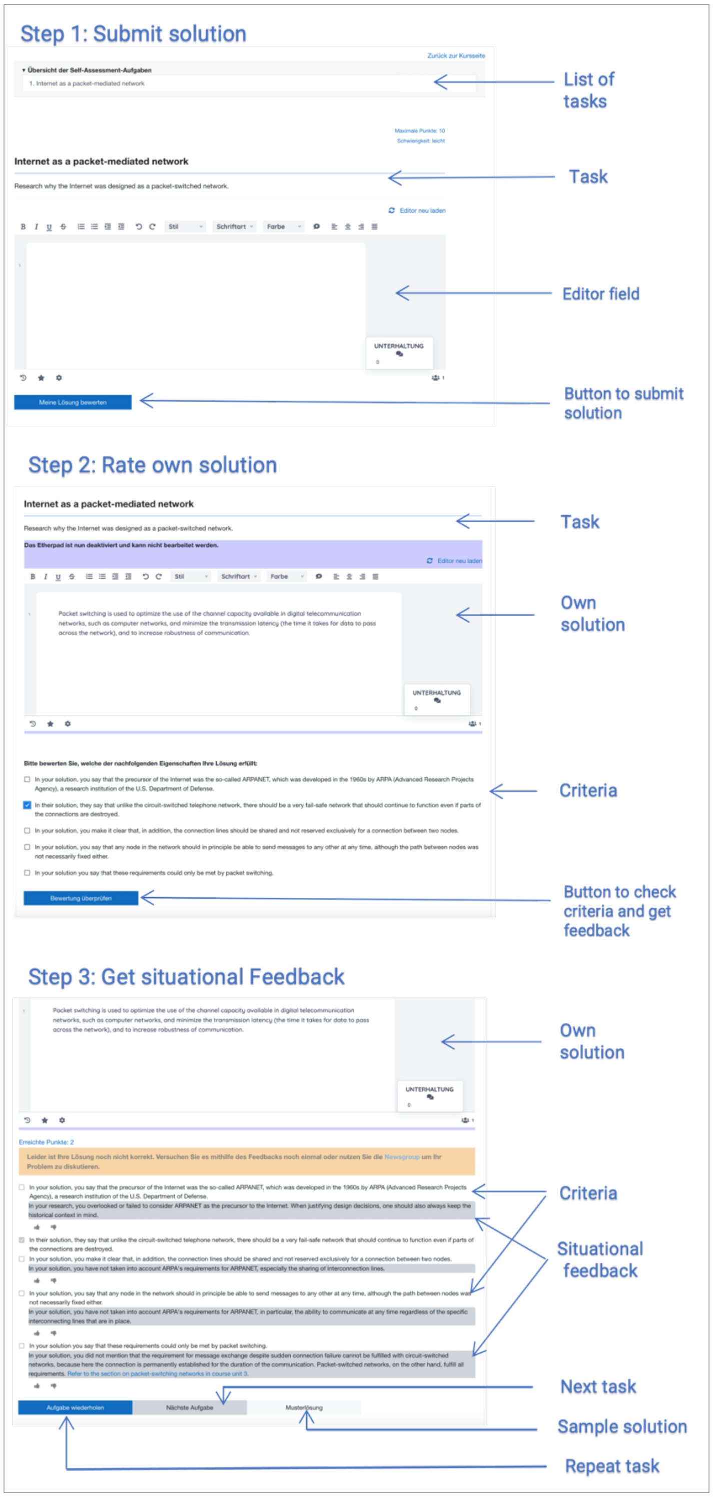
Figure 1 shows the user interface as well as the process of working on a Self-Assessment task in the enhanced SAFRAN plugin. In the first step, a student works on the task and submits his solution. In a second step, the student evaluates his solution by rating whether each indicated criterion is fulfilled by the submitted solution (checked) or not (unchecked). In the final step, the student receives feedback appropriate to his or her Self-Assessment of his or her solution. Here, the student has also the possibility to get access to the sample solution. Now, the editing process of the task can either be repeated to improve the solution based on the feedback or the provided sample solution, or the student may switch to another task, perform other activities within the course, or finish the task.

However, it is not yet sufficiently known how students actually process tasks of such type during the learning process. In order to provide students with usable and beneficial Self-Assessment tasks in a virtual learning environment, it is necessary to study how students deal with solving such tasks during their learning process. In this context, a closer look at the use of feedback and sample solutions is also relevant, as these are among the most important building blocks of the task-solving process.

To address these gaps, this study will answer the following two research questions:

RQ 1: How do students work with Self-Assessment Tasks in SAFRAN and what differences can be observed in the way they process them?

RQ 2: What behavioral patterns can be observed in the use of feedback and sample solutions by students when processing Self-Assessment tasks in SAFRAN?



Internet as a packet-mediated network

Research why the Internet was designed as a packet-switched network.

Das Etherpad ist nun deaktiviert und kann nicht bearbeitet werden.

Packet switching is used to optimize the use of the channel capacity available in digital telecommunication networks, such as computer networks, and minimize the transmission latency (the time it takes for data to pass across the network), and to increase robustness of communication.

Bitte bewerten Sie, welche der nachfolgenden Eigenschaften Ihre Lösung erfüllt:

- ☐ In your solution, you say that the precursor of the Internet was the so-called ARPANET, which was developed in the 1960s by ARPA (Advanced Research Projects Agency), a research institution of the U.S. Department of Defense.
- ☑ In their solution, they say that unlike the circuit-switched telephone network, there should be a very fail-safe network that should continue to function even if parts of the connections are destroyed.
- ☐ In your solution, you make it clear that, in addition, the connection lines should be shared and not reserved exclusively for a connection between two nodes.
- ☐ In your solution, you say that any node in the network should in principle be able to send messages to any other at any time, although the path between nodes was not necessarily fixed either.
- ☐ In your solution you say that these requirements could only be met by packet switching.

Leider ist Ihre Lösung noch nicht korrekt. Versuchen Sie es mithilfe des Feedbacks noch einmal oder nutzen Sie die Newsgroup um Ihr Problem zu diskutieren.

- ☐ In your solution, you say that the precursor of the Internet was the so-called ARPANET, which was developed in the 1960s by ARPA (Advanced Research Projects Agency), a research institution of the U.S. Department of Defense.
  In your research, you overlooked or failed to consider ARPANET as the precursor to the Internet. When justifying design decisions, one should also always keep the historical context in mind.
- ☑ In their solution, they say that unlike the circuit-switched telephone network, there should be a very fail-safe network that should continue to function even if parts of the connections are destroyed.
- ☐ In your solution, you make it clear that, in addition, the connection lines should be shared and not reserved exclusively for a connection between two nodes.
  In your solution, you have not taken into account ARPA's requirements for ARPANET, especially the sharing of interconnection lines.
- ☐ In your solution, you say that any node in the network should in principle be able to send messages to any other at any time, although the path between nodes was not necessarily fixed either.
  In your solution, you have not taken into account ARPA's requirements for ARPANET, in particular, the ability to communicate at any time regardless of the specific interconnecting lines that are in place.
- ☐ In your solution you say that these requirements could only be met by packet switching.
  In your solution, you did not mention that the requirement for message exchange despite sudden connection failure cannot be fulfilled with circuit-switched networks, because here the connection is permanently established for the duration of the communication. Packet-switched networks, on the other hand, fulfill all requirements. Refer to the section on packet-switching networks in course unit 3.

Figure 1. Example of a student's interaction with the SAFRAN plugin

2. RELATED WORK

Since the start of digitalization in the educational environment and especially later on due to the transfer of traditional face-to-face teaching to technologically supported distance teaching accelerated by the COVID-19 pandemic [1], recent research has been concerned with the learning behavior of students in virtual learning environments. Research into this area provides information about the learning processes of different students. This knowledge is important to enable and support the successful acquisition of knowledge by students as much as possible [33]. An important point here is that knowledge acquisition, development and use can be identified through the observation and analysis of the handling of tasks [21]. Many studies use good grades as an indication of knowledge acquisition and use [19, 17, 34, 23, 9, 6, 3].

For example, in analyzing the activity logs of 124 participants from three Moodle courses at three different universities, [34] found a significant positive correlation between task completion and final grade. Their study also showed that students, who were very active within the course and had many logged events, received the highest grades. [17] found a similar result when analyzing the handling of Self-Assessment tasks. They found a positive correlation between engagement in the tasks and good performance in the final exam. For this purpose, they examined log data as well as the self-reports of 159 students of an Economic and Business Education university course.

In general, one could assume from this that students, who actively engage with the course and complete assignments, perform well. However, it remains unknown how these tasks were used for learning. For example, if the tasks were mandatory tasks that possessed a deadline. Thus, the completion of tasks was bound to obligatory aspects, such as time, correctness, and quantity. This could distort the picture of how tasks were handled. Thus, in most studies, a strong increase in activity was always observed during or shortly before a deadline [34, 9].

[25] recognized this lack of interpretability and limited their study to pure practice tasks without evaluation. They analyzed the potential relevance and impact of conducting non-evaluative assessments before rated assessments in an online mathematics course at a university. They found that the performance of practice tasks had a positive impact on the chances of passing the subject. However, as the complexity of the tasks increases, the relevance of participation in non-assessment practice tasks also increases. This result is consistent with standard learning theory [21]. [19] also investigated quiz-taking behavior. They analyzed students' interactions in several online quizzes from different courses and with different settings using process mining. Four different behaviors were identified, a standard quiz-taking behavior, a feedback-using behavior (students using feedback from previous attempts), the use of learning materials during the task, and multitasking behavior (performing other learning activities in the course while working on a task).

Thus, it is known that such behavior patterns exist, but little information is available on how students engage with tasks and whether there are differences in usage. Behavioral patterns of feedback and sample solutions use related to task completion are also not considered. However, this is important in order to gain a better understanding of how tasks are used and to provide appropriate learning opportunities for diverse students. Therefore, with this study we try to gain insight into the behavior of students in dealing with tasks and the corresponding feedback as well as sample solutions.

3. METHODS

To identify task processing behavior patterns of students in a real learning environment with Self-Assessment tasks and corresponding feedback as well as sample solutions, the task processing behavior of students will be investigated by means of learning analytics. For this purpose, a time period within the course is chosen where it can be assumed that students are not engaged in exam preparations or settling in within the course. First, the study design as well as the used dataset will be explained, followed by data collection and analysis methods used.

3.1 Study Design and Dataset

For the study, 254 students of a computer science course on operating systems and computer networks were selected who volunteered to use an adaptive Moodle learning environment in winter term (WT) 2022 and agreed to the study by signing the consent form, which was approved in advance by the university's data protection officer. Students were informed about the use and handling of their data. Only anonymized data was used for analysis. Alternative printed and digital learning material was offered to non-participating students.

The course was divided into four course units. In each of these units, course material and exercises, such as multiple choice (23 occurrences), assignments corrected by tutor (30 occurrences) and Self-Assessments (41 occurrences), were provided. Assignments had a deadline and had to be submitted on time, all other exercises could be completed voluntarily and had no restrictions regarding deadline and repeatability. In addition, a usenet forum, recordings of live sessions, and questions for exam preparation were offered.

The Self-Assessment tasks [14] used in the study were evenly distributed over the individual learning units of the course. The level of difficulty of the tasks was determined by the teacher and was on average in the medium range. The number of Self-Assessment criteria ranged from 2 to 7.

The course started on October 1st, 2022 and ended on February 3rd, 2023. The course was completed with a final exam at the end of semester. In order to get an insight into the learning process, a period of eight weeks was chosen in the middle of the course from 17. October 17, 2022 to December 11, 2022. During this period, it was expected that students …:

- have already completed the introduction of the learning environment.
- are aware of the materials and exercises offered in the course.
- are not yet in the exam preparation phase.

Table 1 lists all possible activities that are distinguished during task processing by the SAFRAN plugin and stored in the log database. Thus, the task ID, the number of attempts, the selected criteria with which the student has evaluated his solution, the activity in which the student is, the timestamp, the user ID and the percentage of points achieved are stored. The activities that a student can perform while working on a task are limited by the plugin. Students are generally able to select a task from a list of tasks and thus open it (open_task_from_list). They can write a solution to the task in the editor and submit this solution for Self-Assessment (request_evaluation). They can evaluate their own solution based on criteria and get feedback for this self-evaluation (request_feedback). Afterwards, students can follow feedback links (clicked_on_link), view the sample solution (request_sample_solution), repeat the task (repeat_same_task), call up the next task in the list (open_next_task), or again select a task from the list (open_task_from_list). In addition, data on the feedback rating, a reason for each negative rating and the kind of feedback, were also collected.

Table 1. An overview of stored task-based properties

pre-processed task activities	meaning
questionid	ID of the task
attempt	number of attempts by student for each task
user_error_situation	number and order of selected criteria of a task iteration
state	activities of students within a task including: - cancle_task (go back to course page) - clicked_on_link (clicked on a link in the feedback) - open_next_task (used button to the next task) - open_task_from_list (used task list to choose a task) - repeat_same_task (repeaded the same task) - request_evaluation (handed in solution and started rating) - request_feedback (rated the solution and got feedback) - request_sample_solution (opend the sample solution) - viewed_task_history (looked at their prior solution and solution rating)
datetime	time the activity is called
userid	ID of the user
achived_points_percentage	result of student's last Self-Assessment attempt, compared to the maximum achievable assessment result
feedbackid	ID of feedback
feedback_rating	positive (1) and negative (0) rating of feedback by user
feedback_reason	reason for negative feedback given by students

From these traces of the participants' interaction with the Self-Assessment plugin, the following indicators were created and used:

- Number of attempts by students for each Self-Assessment.
- Number of sessions students have spent in SAFRAN.
- Number of Self-Assessment sessions per user
- Students' processing time for each Self-Assessment session.
- Number of task changes inside a session
- Number of completed tasks
- Number of sample solution calls per Self-Assessment task by students
- Time needed for students to view the sample solution after requesting feedback.
- Average percentage of points achieved on student Self-Assessment attempts
- Sequences of the different states and questions.

3.2 Data Collection & Analysis

Considering the objective of this study, it is an exploratory study using k-means clustering [26] and process mining methods [28] to identify and map students' behavioral patterns when completing Self-Assessment tasks, as well as to identify how they deal with feedback and sample solutions.

The trace data logged by the SAFRAN plugin were extracted from the database, cleaned, and processed for analysis. To determine the optimal number of clusters (groups of students), the with-cluster sum of squares WCSS [32] and the average silhouette measure [20] were used as clustering quality measures (Appendix 1 Fig 6, 7, 8). Process mining is used to identify processes based on the trace data (Leno et al., 2018). This data is thereby analyzed and mapped into a process model by using the sequence of events to construct the graph. Here, nodes correspond to activities, arcs represent relationships, and each node and arc is annotated with the corresponding frequency. The pm4py [10] Python process mining library was used to construct the process map.

In addition, to understand the relationship and significance of the use of sample solutions during task processing, the Pearson correlation was applied [11]. The Pearson correlation coefficient indicates a linear relation between two indicators and denotes the confidence interval at which the coefficient is significant. It ranges between −1 to +1 and values closer to −1 and +1 imply a strong correlation. A negative correlation coefficient implies a decrease in one indicator would result in an increase in another indicator, and vice versa.

4. RESULTS AND DISCUSSION

4.1 General results on task processing, use of feedback and sample solutions by students

From a total of 254 observed participants, 144 dealt with Self-Assessment tasks at least once during the selected period. Thereby, the 41 available Self-Assessments were processed 1496 times.

During the study period, a student worked on an average of 11 tasks (SD=8.8). The average time needed to complete one task was 3.3 minutes (SD=3.2). Based on their own assessment, students achieved an average correctness of 80% (SD=24.2) of their solution. The tasks were completed in an average of four (SD=3.9) independent activity periods (sessions). A session lasted an average of 34 minutes (SD=34). Within a session, students switched tasks an average of 9.5 times (SD=9).

Tasks were repeated an average of 1.3 times independent of sessions (SD=2.2). The average time from requesting feedback to check one's own solution to requesting the sample solution was 15 seconds (SD=4).

The sample solution was viewed a total of 570 times by students, whereas the percentage distribution of views for sample solutions to completed tasks was M=38%. To understand the relationship and significance of the use of sample solutions during task processing (Fig. 2), a correlation between the achieved relative score and the sample solution calls per task was found to have a strong significant negative correlation (-0.70, p<0.0001%). Thus, it could be assumed that a student with a low achieved task score is more likely to request the sample solution than a student with a high score.

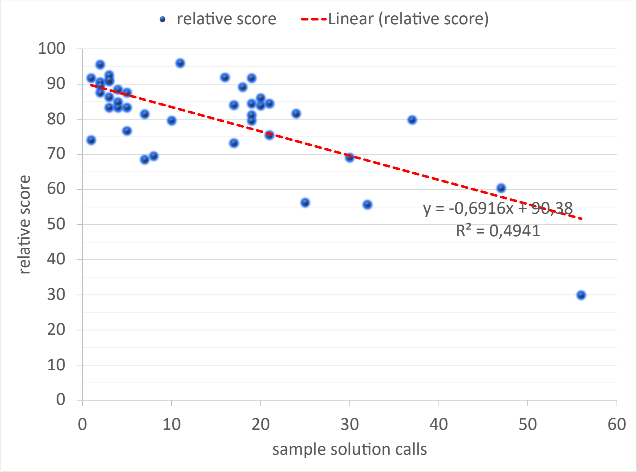


Figure 2. Students call of sample solution

4.2 Differences in students' Self-Assessment task processing

In general, three process clusters could be identified from the data. These clusters show respective process flows that the students performed during their Self-Assessment task processing sessions.

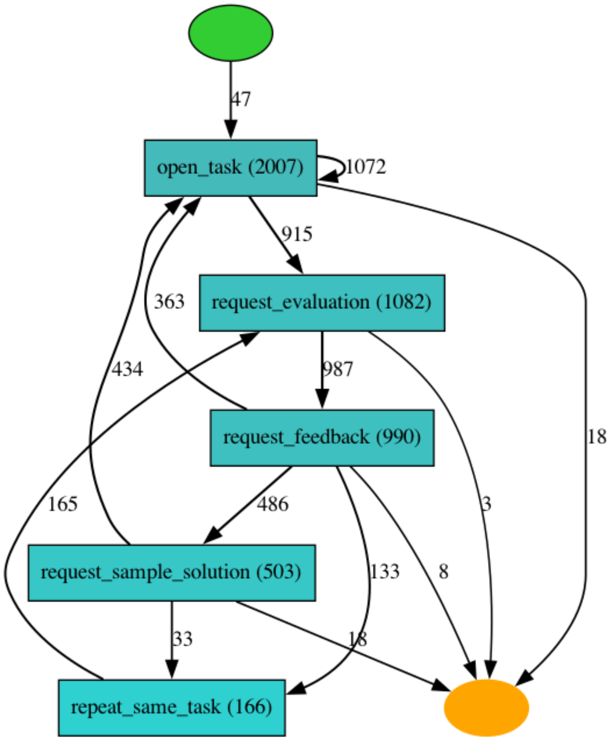


Figure 3. Process tree of the first cluster

Cluster 1 (N=47) shows an intensive processing of tasks (Fig.3). Students in this cluster worked on the tasks for a longer period of time (M=43 min., SD= 26.1). Thereby, students in this cluster have worked on 11 tasks on average (SD 7.1), where solutions were partially or completely correct. Time on task is approximately 3.9 minutes. Based on their own assessment, students in this cluster achieved an average correctness of 81.5% (SD=23) of their solution and repeated a task an average of 2.5 times (SD=2.9). The sample solution was accessed 503 times by these students (50.1%). Students in this cluster seem interested in constructing their own correct solution (mastery of task) as evidenced by on average 2.5 repetitions leading to a relatively high correctness level. Students employed multiple pathways (i.e. activity sequences) mirroring different learning strategies, e.g., using feedback for improvement or using sample solution to identify and to correct deficits. Most frequently, students open a task, request a self-evaluation, request feedback, request sample solution, and repeat the task. The less frequently used pathway is students open a task, request a self-evaluation, request feedback, and repeat the task. As indicated by the relatively high correctness level, the Self-Assessment task type supports students with different learning strategies.

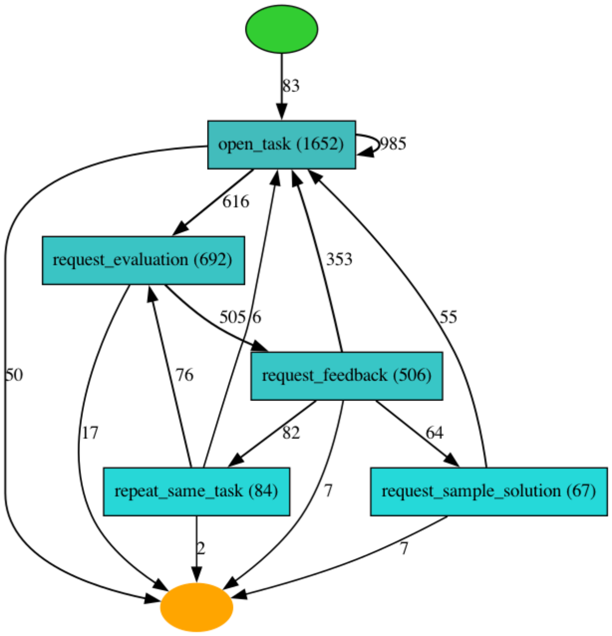


Figure 4. Process tree of the second cluster

The second cluster (N=83) shows a less intensive task processing (Fig.4). Students in cluster 2 worked on average about M=17.6 minutes (SD=7.9) in one session. Thereby, students in this cluster worked on 8 tasks partially to completely on average (SD=8.4). Time on task is approximately 2.2 minutes. According to their own assessment, students in this cluster achieved an average correctness of 80.2% (SD=25.3) of their solution and repeated a task an average of 0.8 times (SD=1.6). The sample solution was accessed 67 times by these students (13.2%).

Students typically opened a task, request evaluation, request feedback, request sample solution, followed by a repeat same task or quit or open next task. The less frequently pathway is open task and next task/quit. Students in this cluster could either be successful learners who reached a high enough score with, at maximum, one iteration in shorter time than students in cluster 1. Or they could exhibit superficial self-evaluation behavior by selecting all criteria as fulfilled and thereby reaching an inappropriate high score.

Similar results can be observed in the log data of mandatory assignments (reviewed by correctors). This suggests that the probability that students exhibit superficial self-evaluation behavior is rather low, but cannot be excluded (cluster 1 (N=28) = 76% assignment correctness, cluster 2 (N=41) = 72% assignment correctness, cluster 3 (N=4) = 73% assignment correctness).

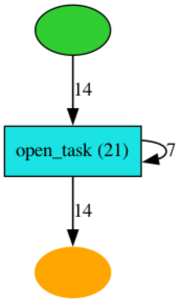


Figure 5. Process tree of the third cluster

The third cluster (N=14) exhibits an incomplete task completion process (Fig.5). Students in this cluster never finished tasks and were limited to just opening different tasks. In doing so, they were active in a session for about M=8 seconds on average, SD=0.2, and switched between tasks about 2 times (SD=16).

Students in this cluster typically opened a task and then opened another task or quitted. Thus, students seem to be interested in getting a quick impression of the task, and may not be interested of working on the task. Students in this cluster often disappear from the course without further activity in the LMS.

5. CONCLUSION

In this study, a process-oriented approach was used to analyze the behavior of the students when dealing with Self-Assessment tasks and to identify differences (RQ1). In addition, the handling of feedback and sample solutions during the processing of Self-Assessment tasks should be considered more closely (RQ2). For this purpose, the process model from [12] was adopted and the prototypical implementation from [30] was adapted and further developed. The newly created Moodle Activity Plugin SAFRAN was able to offer students a simpler iteration of Self-Assessment task and could also provide additional information about the activities carried out in connection with students' task processing.

In general, three different ways of processing the Self-Assessment tasks in the SAFRAN plugin could be observed. These would be an intensive Self-Assessment task processing, as seen in cluster 1 (Fig.3), as well as a moderate task processing of Self-Assessment tasks as seen in cluster 2 (Fig. 4). The process flows in clusters 1 and 2 differ only minimally, since the process steps for solving a task are largely specified by the SAFRAN plugin. They differ only in the proportionality of the activities (pathways). Thus, in cluster 1 the proportionality is approximately uniformly distributed across all activity options, whereas students of cluster 2 predominantly choose one specific process per session and keep it. Students in clusters 1 and 2 differ significantly in the average time required per session and task, in the number of times a sample solution is requested, and in the average number of repetitions per task. All these values are higher in cluster 1 than in cluster 2. Cluster 3, however, is very different from the other two in that it shows only a minimal task process. This mostly consists of just opening the task. Students in this cluster could be classified as task browsers. Similar behavioral patterns could be detected by [12, 19].

Based on the average Self-Assessment scores achieved, both Cluster 1 and Cluster 2 appear to be beneficial behaviors. However, this is not true for Cluster 3, which has an unfavorable task processing pattern. Here, the system would have to adaptively respond to students with such task processing patterns and motivate them to perform tasks in a favorable manner.

Regarding the use of feedback and sample solutions while processing Self-Assessment tasks, it could be observed that students use both feedback and sample solutions. Sample solutions are generally accessed relatively often, but this can strongly vary per task. It seems that the achieved score has an influence on the use and the necessity of a sample solution. The lower the achieved score, the more often students request a sample solution. Based on the average size of the feedback texts (approx. 33 words) and the amount of time students spend on the evaluation page (approx. 15 seconds) that provides feedback, it can be assumed that the feedback has been fully read and taken into account. The fact that students still call up the sample solution despite having received feedback could be due to the fact that for some students the feedback does not seem to be sufficient, so that efforts are apparently made to obtain the information that is still missing with the help of the sample solution. Such behavior indicates that the information content of existing feedback is not always sufficient and should be enriched with additional information. However, this should be adaptively adjusted to the specific needs of students according to their diversity-related characteristics.

5.1 Limitations

The study was only conducted for one course and one subject area, limiting the generalizability of the results. Further studies should therefore also include other subject areas. Since the study period is relatively small compared to the entire semester, no change in learning behavior over time was examined. Also, beneficial task processing patterns were be determined based on Self-Assessment results and the intended Self-Assessment task processing only. However, since we do not yet have results from the final exam, the benefits of the identified task processing patterns cannot be confirmed yet.

In addition, there were also students who did not show any activities of Self-Assessment task processing. However, this does not necessarily imply drop-out, since they could have done other activities in the course, such as reading or mandatory assignments.

6. ACKNOWLEDGMENTS

This research was supported by the Center of Advanced Technology for Assisted Learning and Predictive Analytics (CATALPA) of the FernUniversität in Hagen, Germany.

7. REFERENCES

[1] Adedoyin, O.B. and Soykan, E. 2023. Covid-19 pandemic and online learning: The challenges and opportunities. *Interactive Learning Environments*. (2023), 31:2, 863–875.

[2] Andrade, H.L. 2019. A critical review of research on Student Self-Assessment. *Frontiers in Education*. 4:87, (2019).

[3] Andreswari, R., Fauzi, R., Valensia, L. and Chanifah, S. 2022. Conformance analysis of student activities to evaluate implementation of outcome-based education in early of pandemic using process mining. *SHS Web of Conferences*. 139, (2022), 03018.

[4] Bloom, B.S. 1956. Taxonomy of educational objectives: The classification of educational goals by a committee of College and University Examiners. Longmans.

[5] Brown, G.T. and Harris, L.R. 2013. Student Self-Assessment. *SAGE Handbook of Research on Classroom Assessment*. (2013), 367–393.

[6] Chung, C.-Y. and Hsiao, I.-H. 2020. Investigating patterns of study persistence on self-assessment platform of programming problem-solving. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. (2020), 162–168.

[7] Coussement, K., Phan, M., De Caigny, A., Benoit, D.F. and Raes, A. 2020. Predicting student dropout in subscription-based online learning environments: The beneficial impact of the logit leaf model. *Decision Support Systems*. 135, (2020), 113325.

[8] Csapó, B. and Molnár, G. 2019. Online diagnostic assessment in support of personalized teaching and learning: The edia system. *Frontiers in Psychology*. 10:1522, (2019).

[9] Fouh, E., Breakiron, D.A., Hamouda, S., Farghally, M.F. and Shaffer, C.A. 2014. Exploring students learning behavior with an interactive etextbook in computer science courses. *Computers in Human Behavior*. 41, (2014), 478–485.

[10] Fraunhofer Institute for Applied Information Technology (FIT), process mining group 2019. PM4PY. *State-of-the-art-process mining in Python*.

[11] Freund, R.J., Wilson, W.J. and Sa, P. 2006. *Regression analysis: Statistical modeling of a response variable*. Elsevier Academic Press.

[12] Haake, J.M., Seidel, N., Karolyi, H. and Ma, L. 2020. Self-Assessment mit High-Information Feedback. In: Zender, R., Ifenthaler, D., Leonhardt, T. & Schumacher, C. (ed.). *DELFI 2020–Die 18. Fachtagung Bildungstechnologien der Gesellschaft für Informatik e.V.* (Bonn, 2020), Bonn: Gesellschaft für Informatik e.V., 145–150.

[13] Haake, J.M., Kasakowskij, R., Karolyi, H., Burchard, M. and Seidel, N. 2021. Accuracy of self-assessments in higher education. *In: Kienle, A., Harrer, A., Haake, J. M. & Lingnau, A. (ed.), DELFI 2021* (Virtual, 2021), Bonn: Gesellschaft für Informatik e.V., 97–108.

[14] Haake, J.M., Ma, L. and Seidel, N. 2021. Self-Assessment Questions - Operating Systems and Computer Networks. DOI= 10.5281/zenodo.5021350, 2021.

[15] Hattie, J. and Timperley, H. 2007. The power of feedback. *Review of Educational Research*. 77, 1 (2007), 81–112.

[16] Hew, K.F. and Cheung, W.S. 2014. Students’ and instructors’ use of massive open online courses (moocs): Motivations and challenges. *Educational Research Review*. 12, (2014), 45–58.

[17] Ifenthaler, D., Schumacher, C. and Kuzilek, J. 2022. Investigating students' use of self-assessments in higher education using learning analytics. *Journal of Computer Assisted Learning*. (2022), 39(1), 255–268.

[18] Jordan, A., Ross, N., Krauss, S., Baumert, J., Blum, W., Neubrand, M., Löwen, K., Brunner, M. and Kunter, M. 2006. *Klassifikationsschema für Mathematikaufgaben: Dokumentation der Aufgabenkategorisierung im COACTIV-Projekt [Classification scheme for mathematics tasks: Documentation of task categorization in the COACTIV project].* Max-Planck-Institut für Bildungsforschung [Max Planck Institute for Human Development].

[19] Juhaňák, L., Zounek, J. and Rohlíková, L. 2019. Using process mining to analyze students' quiz-taking behavior patterns in a learning management system. *Computers in Human Behavior*. 92, (2019), 496–506.

[20] Kaufman, L. and Rousseeuw, P.J. 2009. *Finding groups in data: An introduction to cluster analysis*. John Wiley & Sons.

[21] Kleinknecht, M. 2019. Aufgaben und Aufgabenkultur. *Zeitschrift für Grundschulforschung*. 12, 1 (2019), 1–14.

[22] Klieme, E. 2018. Unterrichtsqualität [Teaching quality]. *Handbuch Schulpädagogik*. Waxmann. 393–408.

[23] Kokoç, M., Akçapınar, G. and Hasnine, M.N. 2021. Unfolding Students’ Online Assignment Submission Behavioral Patternsusing Temporal Learning Analytics. *Educational Technology & Society*. 24, 1 (2021), 223–235.

[24] Maier, U., Kleinknecht, M., Metz, K., Schymala, M. and Bohl, T. 2010. Entwicklung und Erprobung eines Kategoriensystems für die fächerübergreifende Aufgabenanalyse. *Schulpädagogische Untersuchungen Nürnberg*. 38, (2010).

[25] Martínez-Carrascal, J.A. and Sancho-Vinuesa, T. 2022. Using Process Mining to determine the relevance and impact of performing non-evaluative quizzes before evaluative assessments. *Learning Analytics Summer Institute Spain (LASI Spain)* (Salamanca, June 20–21, 2022), 52–60.

[26] Murphy, K.P. 2012. Machine learning a probabilistic perspective. MIT Press.

[27] Priemer, B., Eilerts, K., Filler, A., Pinkwart, N., Rösken-Winter, B., Tiemann, R. and Zu Belzen, A.U. 2019. A framework to foster problem-solving in STEM and computing education. *Research in Science & Technological Education*. 38, 1 (2019), 105–130.

[28] Romero, C. 2011. Handbook of Educational Data Mining. CRC Press.

[29] Rudian, S. and Pinkwart, N. 2021. Generating adaptive and personalized language learning online courses in Moodle with individual learning paths using templates. *2021 International Conference on Advanced Learning Technologies (ICALT)*. (2021), 53–55.

[30] Steinkohl, K., Burchart, M., Haake, J. M., & Seidel, N. 2021. *Self-assess question type plugin for moodle*, https://github.com/D2L2/qtype_selfassess.

[31] Terhart, E. 1994. SchulKultur Hintergründe, Formen und Implikationen eines schulpädagogischen Trends [SchoolCulture Backgrounds, forms and implications of a school pedagogical trend]. *Zeitschrift für Pädagogik*. 40, 5 (1994), 685–699.

[32] Thorndike, R.L. 1953. Who belongs in the family? *Psychometrika*. 18, 4 (1953), 267–276.

[33] Wei, X., Saab, N. and Admiraal, W. 2021. Assessment of cognitive, behavioral, and Affective Learning Outcomes in massive open online courses: A systematic literature review. *Computers & Education*. 163, (2021), 104097.

[34] Zhang, Y., Ghandour, A. and Shestak, V. 2020. Using learning analytics to predict students performance in Moodle LMS. *International Journal of Emerging Technologies in Learning (iJET)*. 15, 20 (2020), 102.

8. APPENDIX

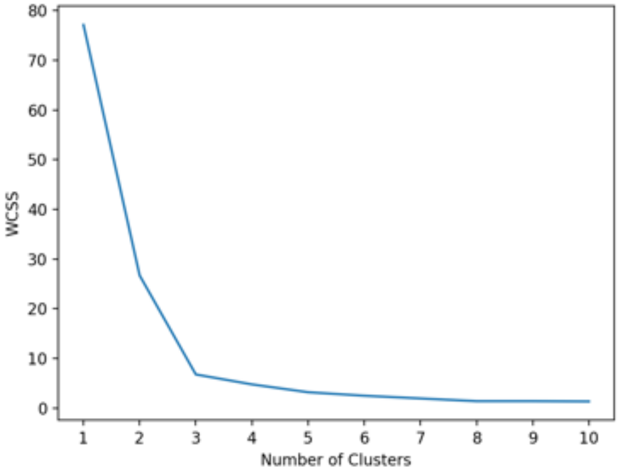


Figure 2. The with-cluster sum of squares (WCSS) result

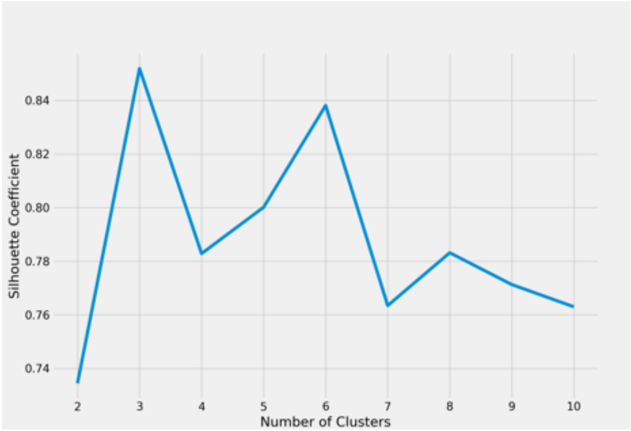


Figure 7. The average silhouette measurement result

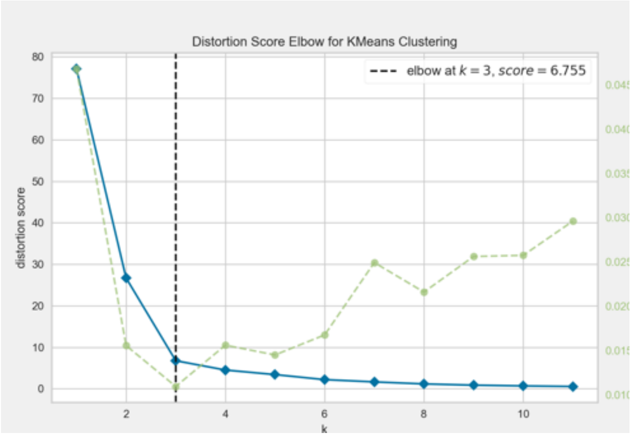


Figure 8. The distortion score elbow result

Using Markov Matrix to Analyze Students' Strategies for Solving Parsons Puzzles

Amruth N. Kumar
Ramapo College of New Jersey
amruth@ramapo.edu

ABSTRACT

Is there a pattern in how students solve Parsons puzzles? Is there a difference between the puzzle-solving strategies of C++ and Java students? We used Markov transition matrix to answer these questions. We analyzed the solutions of introductory programming students solving Parsons puzzles involving `if-else` statements and `while` loops in C++ and Java from fall 2016 to fall 2020. We present the results of our analysis qualitatively as heat maps and quantitatively using descriptive statistics.

We found that most students solved the puzzles in the order in which lines appeared in the correct solution. Counter-intuitively, we found this pattern even in the solutions of the puzzles involving nested `if-else` statements, multiple `while` loops and nested `while` loops. Students who solved the puzzles with the fewest actions acted upon fewer lines out of order, i.e., not in the order in which they appear in the final solution. Whenever we found a statistically significant difference between C++ and Java solutions, C++ solutions involved fewer out-of-order and redundant actions than Java solutions. We discuss the implications of these results for the use of Parsons puzzles as a tool for teaching introductory programming.

Keywords

Parsons puzzles, Puzzle-Solving Strategy, C++, Java, Markov matrix.

1. INTRODUCTION

In a Parsons puzzle [21], first proposed as an engaging way to learn programming, the student is given a program in scrambled order and asked to reassemble it in its correct order. The puzzle may also contain distracters, which are incorrect variants of lines in the puzzle that are meant to be discarded. Parsons puzzles have gained popularity - scores on Parsons puzzles were found to correlate with scores on code-writing exercises [2]. Solving Parsons puzzles was found to take significantly less time than fixing errors in code or writing equivalent code, but resulted in the same learning performance and retention [6]. In electronic books, students preferred solving Parsons puzzles to answering multiple choice questions or writing code [5]. Researchers have placed Parsons puzzles in a hierarchy of programming skills alongside code-tracing [19], and have proposed using it to scaffold software design process [9]. Software to administer Parsons puzzles have been developed for Turbo Pascal [21], Python (e.g., [1,11,12]) and C++/Java/C# [15].

The focus in Parsons puzzles research lately has been on how students solve them and what does/does not help students solve them better, e.g., the patterns in how students solve the puzzles [10,14]; that subgoal labels help students solve puzzles better [20]; that adaptive practice of Parsons puzzles is just as effective as writing code [4, 7]; that students are twice as likely to complete adaptive puzzles than non-adaptive ones [4]; but, motivational supports [16] and the use of mnemonic variable names [13] do not help students solve puzzles more efficiently. Yet, the effectiveness of Parsons puzzles as a tool for learning programming remains unresolved due to lack of replicated research [3] or contradictory results that found no correlation between Parsons puzzles and code-tracing / code-writing exercises [18].

Another focus of research has been on the strategies used by students to solve Parsons puzzles. Each Parsons puzzle typically has only one correct solution. So, the correct solution, i.e., the final reassembled program will be the same for all the students. But, the order in which students go about assembling the lines of code will vary among students. This order reflects their puzzle-solving strategy.

One study found that novice students solved puzzles by focusing on indentation of individual lines or their syntax [8] when lines were presented with indentation. Another study [10] found that some students used "linear" order, i.e., the order in which scrambled lines were provided. But, the study also observed backtracking and looping behavior, which were unproductive. Experts were found to use top-down strategy to solve Parsons puzzles in a study [11]. Students were found to use statement-level semantics more than control-flow semantics to solve puzzles in a recent study [22]. Another study reports that students found the final few steps of the solution to be more challenging [24].

These studies have used various techniques to identify the puzzle-solving strategy of students: think-aloud protocol [8, 11], a state-transition diagram of puzzle-solving states and student transitions [10], edit distance trails and k-means clustering [24] and application of BNF grammar rules to student logs [17]. Think-aloud protocols are gold standard for qualitative research, but they do not scale with the number of participants. State transition diagrams can grow intractable in size with combinatorially explosive number of states in all but very small puzzles, making it hard to find puzzle-solving patterns with the approach. Edit-distance trails [24] lose line-specific information in the puzzles and are better suited for revealing the rate at which students make progress towards the final solution. BNF grammars are suitable for verifying whether a student used a specific puzzle-solving strategy, not for finding the student's strategy.

A. N. Kumar. Using markov matrix to analyze students' strategies for solving parsons puzzles. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 342–350, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115679

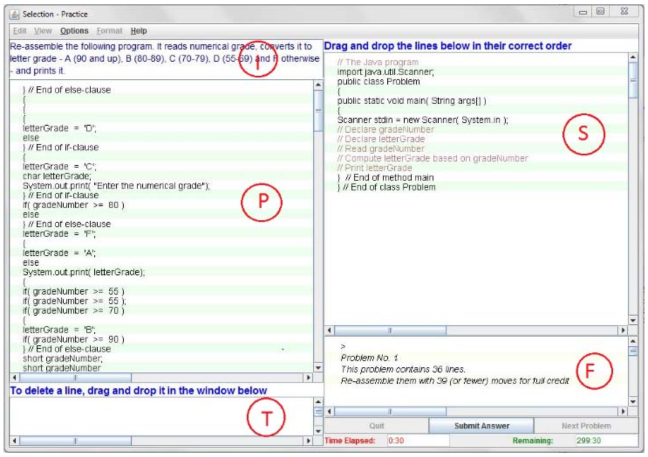
In contrast, a first order Markov transition matrix (not to be mistaken for Hidden Markov Models) can be used to find patterns in time-series data. The matrix has dimensions determined by the number of lines in a puzzle, and not the number of states or students. So, it is scalable with the number of students. We used it to analyze the data collected from the puzzle-solving sessions of students to find patterns or strategies. The research questions for our analysis were:

1. RQ1: Is there a pattern in how students solve the puzzles? Answer to this question may help shed light on how to improve them to promote learning.
2. RQ2: Is there a difference between the puzzle-solving strategies of C++ and Java students? This question is of interest because of the difference in the programming paradigm typically used in the two languages: imperative-first in C++ versus objects-first in Java, even though both the languages support object-oriented programming.

2. PARSONS PUZZLE INTERFACE

For this study, we used the data collected by epplets (epplets.org) [15], a suite of tutors on Parsons puzzles. The user interface of the tutors is shown in Figure 1. The problem statement is displayed in the instruction panel (I). The code for the problem is presented in the problem panel (P), both scrambled and unindented. The solution is assembled in the Solution panel (S). Distracters are deleted when dragged into the Trash panel (T). Feedback is provided for incorrect actions in the Feedback panel (F). The student has the following **action**s available for solving the puzzle:

- **Insert:** Drag a line of code from the Problem panel (P) or the Trash panel (T) to the Solution panel (S) and drop it anywhere in S;
- **Delete:** Drag a line of code from the Problem panel (P) or the Solution panel (S) to the Trash panel (T);
- **Reorder** a line of code in the Solution panel (S) by moving it up or down by one or more lines;
- **Undo:** Return a line from either the Solution panel (S) or the Trash panel (T) back to the Problem panel (P) - the line is placed back in its original scrambled order in the Problem panel (P);



Re-assemble the following program. It reads numerical grade, converts it to letter grade - A (90 and up), B (80-89), C (70-79), D (55-69) and F otherwise - and prints it.

```
} // End of else-clause
{
{
{
letterGrade = 'D';
else
} // End of if-clause
{
letterGrade = 'C';
char letterGrade;
System.out.print( "Enter the numerical grade");
} // End of if-clause
if( gradeNumber >= 80 )
else
} // End of else-clause
letterGrade = 'F';
{
letterGrade = 'A';
else
System.out.print( letterGrade);
{
if( gradeNumber >= 55 )
if( gradeNumber >= 55 );
if( gradeNumber >= 70 )
{
letterGrade = 'B';
if( gradeNumber >= 90 )
} // End of else-clause
short gradeNumber;
short gradeNumber
```

To delete a line, drag and drop it in the window below

Drag and drop the lines below in their correct order

```
// The Java program
import java.util.Scanner;
public class Problem
{
public static void main( String args[] )
{
Scanner stdin = new Scanner( System.in );
// Declare gradeNumber
// Declare letterGrade
// Read gradeNumber
// Compute letterGrade based on gradeNumber
// Print letterGrade
} // End of method main
} // End of class Problem
```

>
Problem No. 1
This problem contains 36 lines.
Re-assemble them with 39 (or fewer) moves for full credit

Figure 1. User Interface of Epplets [15]

In addition, students could indent/outdent lines of code in the Solution panel (S) to improve the readability of the program. But, these actions were not counted in our analysis since indentation does not affect the semantics of C++ and Java programs.

The tutors do not provide any feedback while the student is solving the puzzle. If the student attempts to submit an incomplete solution before moving all the lines out of the panel P, the tutors direct the student to properly place all the lines before submitting their solution. Once a complete solution is submitted, the tutors repeatedly highlight the next line in the solution that is not in its correct location. The tutors either suggest how the line should be moved or point out the line of code that should replace it. The tutors provide such feedback until the solution is correct. The actions taken by the student in response to the feedback become part of the student's solution sequence.

3. MARKOV TRANSITION MATRIX

The tutors report the order in which students solve a Parsons puzzle as a sequence of <`line`, `action`> pairs, `line` referring to line number in the correct solution of the code and `action` referring to the action applied to that line of code. We will refer to this sequence of pairs as action sequence. From a student's action sequence, we can extract the order in which the student acted upon the lines of the puzzle by considering only the first tuple in each pair.

For example, consider a four-line Parsons puzzle with no distracters. The four lines are provided scrambled in panel P (Figure 1). We will refer to these lines by their location in the correct solution, e.g., line 3 is the line that should appear third in the correct solution, although it may be in any order in panel P. Suppose a student solves the puzzle using the following actions:

1. Drags line 3 from panel P to S;
2. Drags line 1 from P to S and drops it after line 3;
3. Moves line 3 after line 1 in S;
4. Drags line 2 from P to S and drops it after line 3;
5. Drags line 4 from P to S and drops it after line 2; and
6. Moves line 2 up so that it appears between lines 1 and 2.

The corresponding action sequence is

1. <3, Insert>
2. <1, Insert>
3. <3, Reorder>
4. <2, Insert>
5. <4, Insert>
6. <2, Reorder>.

From this action sequence, we extract the order in which the student acted upon the lines of the puzzle as 3-1-3-2-4-2. Finally, we use this order of lines to build a Markov transition matrix [25].

In a Markov transition matrix, the rows and columns are line numbers in the program, followed by distracters in the puzzle. In addition, the matrix contains a first row for the start state S before attempting the puzzle and a last column for the end state E after completely solving the puzzle. So, Markov matrix is an n X n matrix where n = number of lines + number of distracters + 1.

We will use M as the abbreviation for Markov transition matrix and $M_{i,j}$ to denote the element of the matrix on row i and column j. Initially, all the elements $M_{i,j} = 0$. If a student applies an action to line j after applying an action to line i, $M_{i,j}$ is incremented by 1.

As an illustration, consider a puzzle containing 4 lines of code that are provided to the student scrambled. The left side of Figure 2 shows the Markov transition matrix of a student who applies actions to lines in the following order: 4-1-2-1-3-4. Since the first line acted on by the student is 4, $M_{S,4} = 1$. Thereafter, the matrix entries that are set to 1 are $M_{4,1}$, $M_{1,2}$, $M_{2,1}$, $M_{1,3}$, $M_{3,4}$ and finally, $M_{4,E}$ since 4 is the last line to be acted upon. The right side of the figure shows the matrix for a student who applies actions to lines in the following order: 1-3-2-2-3-2-4-1. In particular, note that the student acts upon line 2 after line 3 twice – hence, $M_{3,2} = 2$. The student applies back-to-back actions to line 2, e.g., inserts line 2 into the solution, and immediately reorders it in the solution – hence, $M_{2,2} = 1$. The last line acted upon is line 1 – hence, $M_{1,E} = 1$.

For our analysis, we combined the Markov matrices of all the student solutions into a single transition matrix, such that:

$$M_{i,j} = \sum a_{i,j} / s$$

$\sum a_{i,j}$ is the sum of all the actions on line j after line i in all the student solutions;

s is the number of student solutions, i.e., the number of times students solved the puzzle.

	1	2	3	4	E			1	2	3	4	E
S				1			S	1				
1		1	1				1			1		1
2	1						2		1	1	1	
3				1			3		2			
4	1				1		4	1				

Figure 2. Markov Transition Matrices for solution sequences 4-1-2-1-3-4 and 1-3-2-2-3-2-4-1 for a puzzle containing 4 lines of code

So, $M_{i,j}$ is the number of actions on line j after line i per student solution. If all the students applied exactly one action to each line in each solution, $0 \leq M_{i,j} \leq 1$.

Since the puzzles also included two distracters D1 and D2, we added rows and columns in the matrix for D1 and D2 after those for all the lines in the puzzle. $M_{i,D1}$ refers to students acting on the first distracter D1 after line i. In the matrix:

- If each student applies exactly one action to each line of code, the sum of all the entries in a row / column is 1. But, since a student may apply more than one action to a line of code (e.g., insert into the solution, reorder within the solution), the sum of each row / column is at least 1.
- The larger the value of $M_{i,j}$, the larger the number of times students applied an action to line j after line i.
- A puzzle assembled in the correct order of lines, i.e., line 1 in the solution is inserted first ($M_{S,1}$), line 2 in the solution is inserted next ($M_{1,2}$), and so on, will appear as entries in all the diagonal elements of the matrix from top left to bottom right.
- When the solutions of all the students are combined in a matrix, each widely used puzzle-solving strategy produces a distinct pattern in the matrix: entries between frame elements are large in frame-first strategy and most of the elements are non-zero and small in a random strategy.

4. DATA COLLECTION AND ANALYSIS

For this study, we analyzed the data collected online by two Parsons puzzle tutors called epplets (epplets.org) [15] on `if-else` statements and `while` loops. The tutors were used by introductory programming students as after-class assignments in high schools, community colleges and baccalaureate institutions during fall 2016 – fall 2020 as shown in Table 1. Some schools used the tutors for C++ and others for Java – so, the two sets of users were mutually exclusive. C++ and Java versions of each puzzle were of exactly the same size. This made it possible to compare the solutions in the two languages. Since the tutor users were introductory programming students, they had little prior programming experience. The demographics of the students using the two tutors are shown in Table 2. Not everyone reported their gender/race/major.

The tutors were set up to randomize the variable names and data types used in the puzzles. They also randomly scrambled code in the problem panel P. Research shows that novice programming students are unduly influenced by the superficial differences resulting from such randomization [31, 32, 33]. This randomization deterred plagiarism since no student saw the same puzzle verbatim more than once and no two students saw the same puzzle verbatim. It also deterred solution-sharing plagiarism schemes that afflict programming tutors [27].

Table 1. Usage of the tutors in fall 2016 – fall 2020

Fall 2016 – Fall 2020	**if-else**		**while loop**	
Type of Institution	**C++**	**Java**	**C++**	**Java**
High Schools	2	11	1	5
Community Colleges	3	1	2	2
Baccalaureate Institutions	4	13	3	11

For our analysis, we considered only those students who solved a puzzle completely and correctly so that we could find patterns among those who successfully solved the puzzle. Only students who consented to their data being used for research purposes were included in the study. Because of these two factors, the N reported in Table 2 is not the same as those reported in subsequent tables. Since the tutors were accessible over the web, students could use the tutors as often as they pleased. If a student used a tutor more than once, we picked the session in which the student had solved the most number of puzzles. In case of a tie between two sessions, we used the data from only the first session.

Table 2. Demographics of the users of the tutors

Fall 2016 – Fall 2020		**if-else**	**while loop**
N		431	203
Gender	Male	264	102
	Female	100	38
Race	Caucasian	194	78
	Asian	91	33
	Other	70	25
Major	Computer Science	170	76
	Engineering	77	25
	Sciences	22	7

We analyzed the data of each puzzle using three-color heat maps and descriptive statistics. A three-color heat map shows zero values in red, maximal values (0.2 and up) in shades of green and intermediate values (0.1 – 0.2) in yellow. For the calculation of descriptive statistics, we eliminated the last two rows and the penultimate two columns in the matrix corresponding to the two distracters – they were not part of the correct solution. The descriptive statistics included:

1. the number of different lines acted upon first (F) by students, i.e., the number of non-zero cells in the first row of the matrix;
2. the number of different lines acted upon last (L) by students, i.e., the number of non-zero cells in the last column of the matrix;

3. the percentage of matrix cells (C) that are non-diagonal and non-zero; and
4. the sum of the values (V) of non-diagonal non-zero matrix cells expressed as a percentage of the sum of the values of all non-zero cells.
5. The mean of diagonal elements (μ_d).

Note that the greater the values of F and L, the more varied the solutions. The larger the value of C, the more the lines that were acted upon out of order, i.e., not in the order in which they appear in the final solution. The larger the value V, the more the redundant actions and hence, the less efficient the solutions.

A puzzle with n lines can be solved with n actions. For the purposes of analysis, we considered as minimal solvers, students who solved a puzzle with no more than 1.1n actions, i.e., with no more than 10% redundant actions. Minimal solvers were a subset of all the solvers of a puzzle. We analyzed the data of each puzzle, both for non-minimal and minimal solvers. We computed the statistical significance of the difference between two groups (e.g., non-minimal versus minimal solvers) by using paired sample t-test in which the corresponding values $M_{i,j}$ (the element of the Markov matrix on row i and column j) of the two groups were paired.

5. RESULTS

5.1 if-else puzzles

The first puzzle solved by the students was on a program that read two numbers and printed the smaller of the two numbers. The puzzle contained 14 lines of code and 2 distracters.

Figure 3 shows the heat map of C++ solutions: for non-minimal solutions on the left and minimal solutions on the right. In the heat maps, the last two rows and the penultimate two columns correspond to distracters. Note the following in Figure 3:

1. A majority of both non-minimal and minimal solvers assembled the puzzle in the order in which the lines appeared in the correct program. So, the largest values are all along the diagonal – μ_d, the mean of diagonal elements, is 0.61 for non-minimal and 0.81 for minimal solvers. This behavior was much more pronounced among minimal solvers: the diagonal is brighter green and far more non-diagonal cells are red (zero). Paired sample t-test yielded a statistically significant difference between the two groups ($p < 0.001$).
2. Students discarded distracters more often than not at the end of the session – the cells in the last column for the last two rows are green.

Minimal solvers solved the puzzles with no more than 10% unnecessary actions. But, this did not mean, they had to assemble the puzzle in the order in which the lines appeared in the correct program (corresponding to the diagonal from top left to bottom right being green): they could have assembled the program in reverse order, i.e., the last line first and the first line last (corresponding to the diagonal elements from the bottom left to the top right being green) or in random order (non-diagonal elements just as likely to be green as diagonal elements). That a majority of both non-minimal and minimal solvers solved the puzzles in the correct order of the lines in the puzzle is a novel and interesting finding of this study.

Figure 4 shows the heat map of Java solutions: for non-minimal solutions on the left and minimal solutions on the right. We observe the same two patterns in Java as in C++. The difference between non-minimal and minimal Java solutions was again statistically significant ($p < 0.001$).

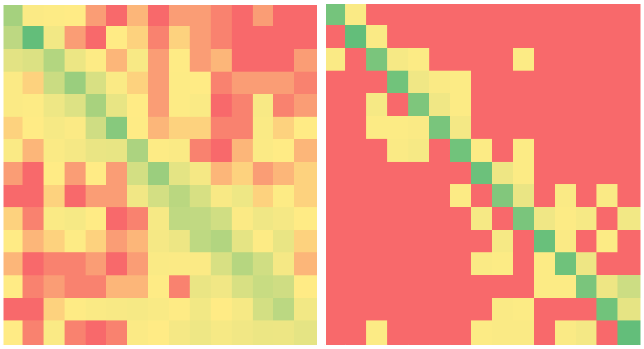


Figure 3. Heat Map of C++ solutions: non-minimal (N=118) on the left and minimal (N=57) on the right

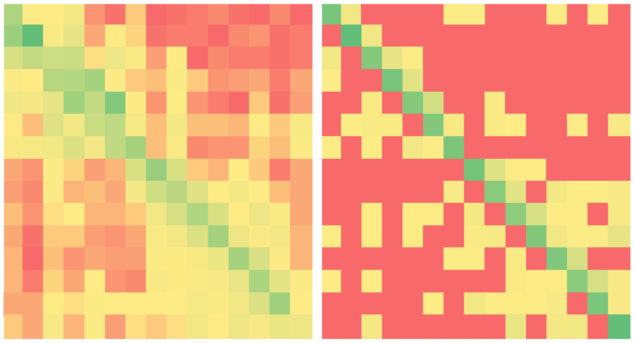


Figure 4. Heat Map of Java solutions: non-minimal (N=237) on the left and minimal (N=66) on the right

Table 3 lists the descriptive statistics for C++ and Java solutions. The table numerically confirms what is hinted at in the heat maps: minimal solutions were less varied, with fewer first lines (F) and last lines (L). In minimal solutions, students acted on fewer lines out of order, e.g., among C++ solvers, non-zero non-diagonal cells (C) were fewer - 24.3% for minimal versus 88.1% for non-minimal C++ solutions. As could be expected, minimal solutions were more efficient with fewer redundant actions, e.g., among Java solvers, the sum of non-zero non-diagonal cells as a percentage of all the non-zero cells (V) was smaller too - 26% for minimal versus 68.8% for non-minimal Java solutions.

Table 3. Descriptive statistics for `if-else` puzzle 1

if-else puzzle 1	C++ (Minimal?)		Java (Minimal?)	
	No	Yes	No	Yes
Sample (N)	118	57	237	66
First line (F)	10	2	12	6
Last line (L)	13	4	14	7
Cells (C)	88.1%	24.3%	96.7%	35.2%
Value (V)	64.4%	15.4%	68.8%	26.0%
Diagonal (μ_d)	0.61	0.81	0.63	0.7

In addition, it is evident from Table 3 that C++ solutions were less varied, had fewer lines assembled out of order and were more efficient than Java solutions. Paired samples t-test yielded a statistically significant difference between non-minimal C++ and Java solutions ($p < 0.001$), but not between minimal C++ and Java solutions.

The second puzzle solved by students was on a program to read numerical grade and print the corresponding letter grade. The program contained four levels of nesting of `if-else` statements. The puzzle contained 34 lines of code and 2 distracters.

The heat maps of C++ solutions are shown in Figure 5 – for non-minimal solutions on the left and minimal solutions on the right. Once again, note that a majority of the students solved the puzzle in the order of the lines in the correct solution. This result is particularly counter-intuitive since the solution contained four levels of nesting of `if-else` statements. Multiple copies of the same line of code (e.g., `else` or braces) were treated as interchangeable by the tutor. Yet, assembling nested `if-else` statements in the order of the lines is no small feat. Balancing the braces of if-clause and else-clause is in itself a difficult task for novice programmers. Yet, a majority of the students chose to reassemble the program in the order in which the lines appear in the correct solution. The difference between non-minimal and minimal C++ solutions was statistically significant ($p < 0.001$).

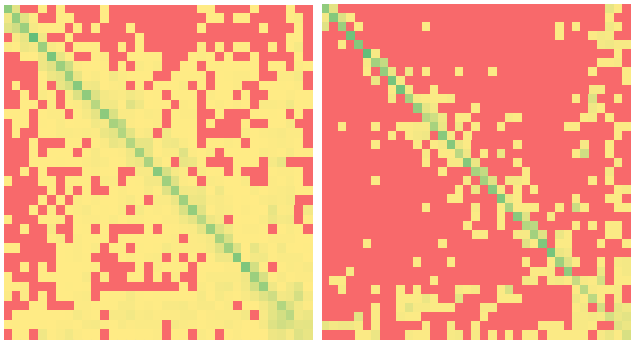


Figure 5. Heat Map of C++ solutions: non-minimal (N=89) on the left and minimal (N=42) on the right

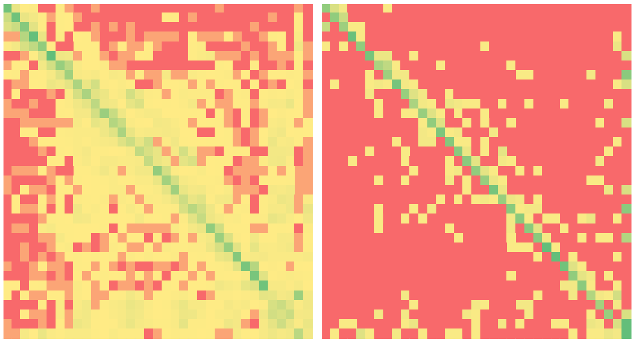


Figure 6. Heat Map of Java solutions: non-minimal (N=154) on the left and minimal (N=28) on the right

Table 4. Descriptive statistics for `if-else` puzzle 2

`if-else` puzzle 2	C++ (Minimal?)		Java (Minimal?)	
	No	Yes	No	Yes
Sample (N)	89	42	154	28
First line (F)	9	2	9	4
Last line (L)	18	12	29	10
Cells (C)	67.4	19.0	78.2	17.4
Value (V)	64.3	38.5	70.4	37.5
Diagonal (μ_d)	0.57	0.62	0.46	0.62

Similarly, the heat maps of Java solutions are shown in Figure 6. Once again, students attempted to solve the puzzle in the order of the lines in the correct solution, minimal solvers much more so. The difference between non-minimal and minimal Java solutions was statistically significant ($p < 0.001$).

The descriptive statistics are shown in Table 4. The difference between C++ and Java was not statistically significant for non-minimal or minimal solutions.

5.2 while loop puzzles

The first puzzle solved by students was on a program to read numbers till the same number appeared back to back. The program printed the first number to appear twice back to back. The puzzle contained 13 lines of code and 2 distracters.

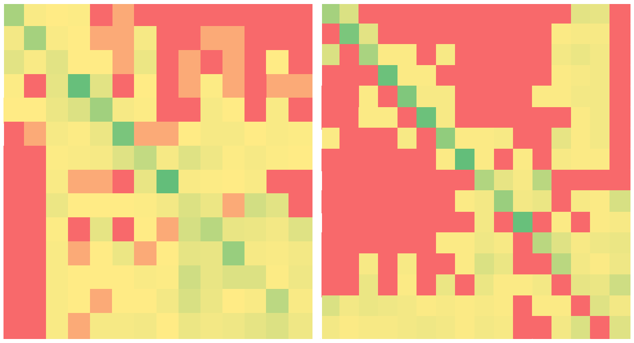


Figure 7. Heat Map of C++ solutions: non-minimal (N=48) on the left and minimal (N=44) on the right

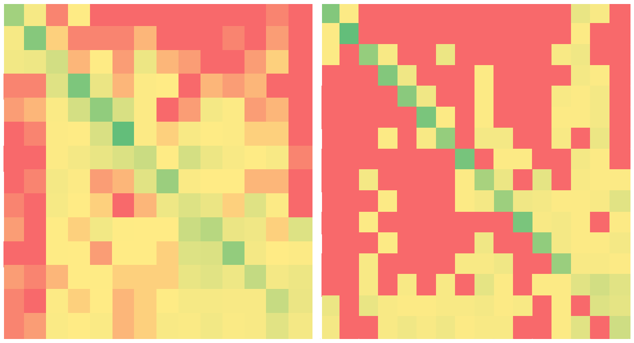


Figure 8. Heat Map of Java solutions: non-minimal (N=78) on the left and minimal (N=42) on the right

Table 5. Descriptive Statistics for `while` puzzle 1

`while` puzzle 1	C++ (Minimal?)		Java (Minimal?)	
	No	Yes	No	Yes
Sample (N)	48	44	78	42
First line (F)	5	2	5	2
Last line (L)	8	5	6	6
Cells (C)	73.1	32.4	80.8	29.1
Value (V)	62.5	33.1	62.9	23.0
Diagonal (μ_d)	0.56	0.6	0.62	0.7

The heat maps of C++ and Java solutions are shown in Figures 7 and 8 respectively. In both of the languages, students solved the puzzle in the correct order of lines, minimal solvers much more so. The difference between non-minimal and minimal solutions was statistically significant for both C++ ($p < 0.001$) and Java ($p < 0.001$). The difference between non-minimal C++ and Java solutions was statistically significant at $p = 0.1$ level, but not the difference between minimal C++ and Java solutions. Table 5 lists the descriptive statistics for all four cases.

The second puzzle solved by the students was on a program to input the face of a card followed by cards in a deck. It prints the number of cards into the deck where it finds the first card, and prints the face of the subsequent card in the deck. The program contained two back-to-back `while` loops. The puzzle contained 22 lines of code and 2 distracters.

The tutors were set up to conduct a controlled experiment on whether using mnemonic variable names affected how efficiently students solved the puzzles [13]. Some schools received a version of the puzzle with mnemonic variable names whereas others received a version with single-character variable names. Since the C++ sample size was larger for the single-character version of the puzzle and Java sample size was larger for the mnemonic version of the puzzle, we used data from those respective versions for comparison.

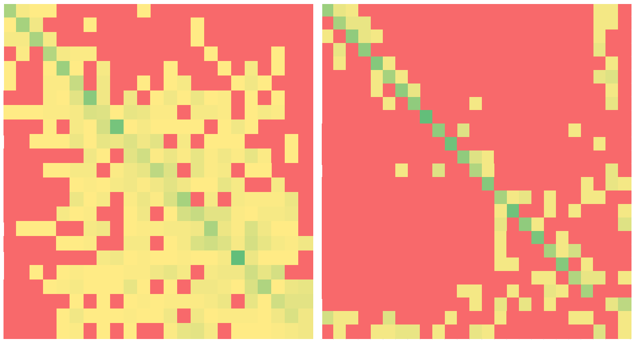


Figure 9. Heat Map of C++ solutions of the single-character version of the puzzle: non-minimal (N=27) on the left and minimal (N=14) on the right

The pattern of students solving puzzles in the correct order of lines is again evident from Figures 9 (of C++ solutions of single-character version of the puzzle) and 10 (of Java solutions of mnemonic version of the puzzle). Quite counter-intuitively, minimal solvers rarely straggled back and forth between the two `while` loops, i.e., picked a line in the first loop followed by a line in the second loop or vice versa. Descriptive statistics are listed in Table 6. The difference between non-minimal and minimal solutions was statistically significant for both of the languages ($p < 0.001$).

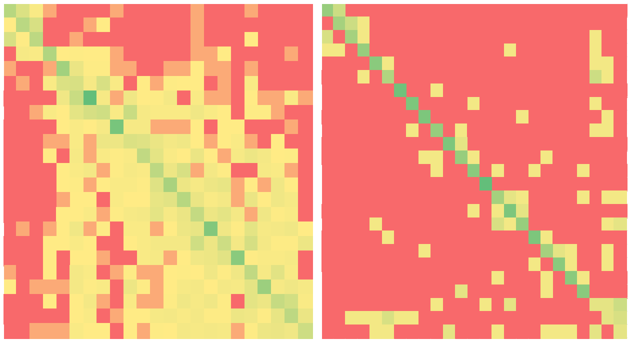


Figure 10. Heat Map of Java solutions of the mnemonic version of the puzzle: non-minimal (N=50) on the left and minimal (N=12) on the right

The third puzzle solved by the students was on a program to repeatedly read a positive number, read additional numbers till its multiple is found, and print the number and its multiple. It did this until 0 or a negative value was input for the first number. The program contained nested `while` loops. The puzzle contained 17 lines of code and 2 distracters.

Figures 11 (C++) and 12 (Java) once again show that a majority of the students solved the puzzles in the correct order of the lines in the solution, even though the puzzle involved nested `while` loops. Nested `while` loops are particularly hard for novice programmers to read or write. So, it is counter-intuitive that students would assemble the lines in the order in which they appear in the correct solution.

Table 6. Descriptive statistics for `while` puzzle 2

while puzzle 2	C++ - single-char (Minimal?)		Java – mnemonic (Minimal?)	
	No	Yes	No	Yes
Sample (N)	27	14	50	12
First line (F)	5	3	7	2
Last line (L)	5	5	7	3
Cells (C)	55.9	11.3	66.4	9.9
Value (V)	68.1	26.3	69.4	25.4
Diagonal (μ_d)	0.58	0.7	0.55	0.71

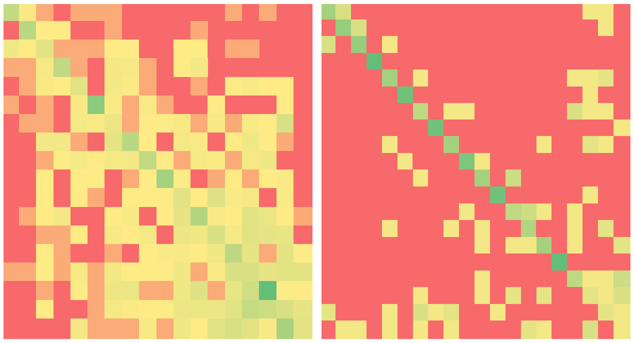


Figure 11. Heat Map of C++ solutions: non-minimal (N=44) on the left and minimal (N=12) on the right

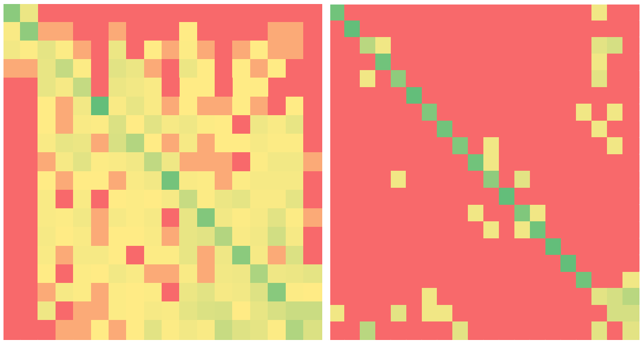


Figure 12. Heat Map of Java solutions: non-minimal (N=48) on the left and minimal (N=11) on the right

The descriptive statistics are shown in Table 7. The difference between non-minimal and minimal solutions was statistically significant for both C++ and Java ($p < 0.001$). The difference between C++ and Java was not statistically significant in either case: non-minimal or minimal solutions. A confounding factor of this comparison is that the number of minimal solvers is small for both C++ and Java.

Table 7. Descriptive statistics for `while` puzzle 3

`while` puzzle 3	C++ (Minimal?)		Java (Minimal?)	
	No	Yes	No	Yes
Sample (N)	44	12	48	11
First line (F)	8	2	2	1
Last line (L)	6	4	6	2
Cells (C)	67.0	11.4	71.6	4.3
Value (V)	67.8	27.9	66.7	7.7

`while` puzzle 3	C++ (Minimal?)		Java (Minimal?)	
	No	Yes	No	Yes
Diagonal (μ_d)	0.58	0.66	0.57	0.85

6. DISCUSSION

We presented the results of analyzing the data of five different puzzles – involving a single `if-else` statement, nested `if-else` statements, a single `while` loop, multiple `while` loops and nested `while` loops. The answer to our research question RQ1 is that in every case, a majority of the students solved the puzzle in the order of the lines of code in the correct solution, as illustrated by the diagonals in heat maps. Students who solved the puzzles with the fewest actions did so by acting upon fewer lines out of order and less often.

An earlier study had used think-aloud protocols to find that experts solved Parsons puzzles [11] by first assembling the majority of the control flow, followed by initialization of variables and handling of corner cases. This was referred to as top-down strategy. In a similar vein, when writing control statements, novices are advised to write the frame of the control statement first and then, proceed to fill in the details [26]. We had hoped to find that at least minimal solvers used such strategies.

Instead, at each step, students seem to have asked themselves "where in the scrambled code can I find the next line of the solution?" instead of "where should the next scrambled line be placed in the solution?" or "how would I write this solution based on top-down thinking and frame-first coding?" They assembled code in the order in which it appears in the program, not the order in which it is written by a programmer who follows top-down decomposition of the problem. *This is the difference between the product and the process.* The order in which code segments are written in a program is dictated by the process of programming and is not necessarily the order in which the code segments eventually appear in the program, i.e., the product of programming. The process is influenced by both semantics (top-down design [11]) and syntax (frame-first programming [26]). Educators want novices to learn the process of programming, not the product, since the product, i.e., the program for a given problem is not unique. Researchers have found that the process used by novices for programming is a better predictor of their course grade than the actual programs written by them [29]. Besides, *product follows process* – the more disciplined the process, the better the programming product. So, for a novice learning to write programs, the focus should be on the process of programming and not the product. Unfortunately, in programming, one cannot learn the process by looking at the product – all the process information is lost by the time a program is completed [30]. So, the fact that a majority of the students solve Parsons puzzles by focusing on the product rather than reconstructing the process of programming does not bode well for Parsons puzzles as a tool for learning programming. Parsons puzzle tutors designed to help students learn programming must actively prompt and scaffold novices to reconstruct the process of programming when solving the puzzles.

Yet, scores on Parsons puzzles were found to *correlate* with scores on code-writing exercises [2]. An explanation for this correlation is that just as they assemble Parsons puzzles, students write programs line by line in the order in which the lines appear in the program, i.e., their process mirrors the product. Writing a program line by line in this manner is difficult because it entails significant cognitive load, e.g., when writing the statements in a nested loop, the programmer must actively keep track of the nested loop, the nesting loop and any variables previously declared in the program. Experts seldom write code in this manner, instead resorting to top-down and frame-first strategies. This *naïve* approach to writing code may explain why attrition in introductory programming courses remains unacceptably high [34, 35]. Configuring Parsons puzzle tutors to proactively enforce top-down and frame-first coding maybe one way to use Parsons puzzles to help students learn effective processes of programming rather than developing their own ineffective processes.

Earlier researchers have reported that C++ students used semantics more than Java students while solving Parsons puzzles [23] and that the learning curve associated with learning object-oriented programing in Java is steeper compared to learning imperative programming in C++ [22]. This may be why we found significant difference between non-minimal C++ and Java solutions on the first problems in both the tutors (our research question RQ2), In both the cases, Java students used more out-of-order and redundant actions to solve Parsons puzzles than C++ students. This finding would benefit from replication in a more controlled environment.

One confounding factor of our study is that the algorithm was provided as comments in the solution panel S (Figure 1) for each program. Students may have followed these comments from top down to assemble the program from the first to the last line. Then again, the presence of comments should have freed students to assemble the different commented sections of the program in an opportunistic manner, not necessarily from the first to the last section. Other researchers have noted that such subgoal labels make it easier for students to solve Parsons puzzles [20], but do not address the influence of comments on how students go about solving Parsons puzzles.

In our analysis, we considered only line numbers in action sequence, the sequence of <`line, action`> pairs. We ignored the information about the action applied to each line, thereby losing some richness of data. For example, $M_{i,i}$ represents back-to-back actions applied to line i. These could be actions that cancel each other out, such as deleting a line followed by undeleting it. In such a case, the two actions could be ignored. Considering the nature of action while creating Markov transition matrix may lead to better results.

In our analysis, we considered only complete and correct solutions. Analyzing incomplete and incorrect solutions may yield patterns in puzzle-solving behavior that unearth common misconceptions among programming students.

We presented Markov matrices as a technique for finding patterns in Parsons puzzle solutions and used heat maps to visualize the results. An added benefit of using Markov matrix is that we can use higher order matrices (obtained by multiplying a Markov matrix by itself) to answer questions such as how quickly after assembling an open brace do students get around to assembling its matching closing brace in a program, a question of interest in frame-first [26] coding.

Knowing how students solve Parsons puzzles can help us understand how they can be improved for that purpose. We hope our discussion in this section contributes towards these efforts. We plan to continue to collect data from additional tutors and analyze the problem-solving patterns used by students in those tutors to see if the same patterns are repeated.

7. ACKNOWLEDGMENTS

Partial support for this work was provided by the National Science Foundation under grants DUE-1502564 and DUE-2142648.

8. REFERENCES

[1] Nick Cheng and Brian Harrington. 2017. The Code Mangler: Evaluating Coding Ability Without Writing any Code. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17). ACM, New York, NY, USA, 123-128. DOI: https://doi.org/10.1145/3017680.3017704.

[2] Paul Denny, Andrew Luxton-Reilly, and Beth Simon. 2008. Evaluating a new Exam Question: Parsons Problems. In Proceedings of the Fourth International Workshop on Computing Education Research (ICER '08). ACM, New York, NY, USA, 113-124. DOI=http://dx.doi.org/10.1145/1404520.1404532.

[3] Yuemeng Du, Andrew Luxton-Reilly, and Paul Denny. 2020. A Review of Research on Parsons Problems. In Proceedings of the Twenty-Second Australasian Computing Education Conference (ACE'20). Association for Computing Machinery, New York, NY, USA, 195–202. DOI:https://doi.org/10.1145/3373165.3373187

[4] Barbara J. Ericson, James D. Foley, and Jochen Rick. 2018. Evaluating the Efficiency and Effectiveness of Adaptive Parsons Problems. In Proceedings of the 2018 ACM Conference on International Computing Education Research (ICER '18). ACM, New York, NY, USA, 60-68. DOI: https://doi.org/10.1145/3230977.3231000

[5] Barbara J. Ericson, Mark J. Guzdial, and Briana B. Morrison. 2015. Analysis of Interactive Features Designed to Enhance Learning in an Ebook. In Proceedings of the eleventh annual International Conference on International Computing Education Research (ICER '15). ACM, New York, NY, USA, 169-178. DOI: https://doi.org/10.1145/2787622.2787731.

[6] Barbara J. Ericson, Lauren E. Margulieux, and Jochen Rick. 2017. Solving Parsons Problems Versus Fixing and Writing Code. In Proceedings of the 17th Koli Calling International Conference on Computing Education Research (Koli Calling '17). ACM, New York, NY, USA, 20-29. DOI: https://doi.org/10.1145/3141880.3141895.

[7] Barbara Ericson, Austin McCall, and Kathryn Cunningham. 2019. Investigating the Affect and Effect of Adaptive Parsons Problems. In Proceedings of the 19th Koli Calling International Conference on Computing Education Research (Koli Calling '19). Association for Computing Machinery, New York, NY, USA, Article 6, 1–10. DOI:https://doi.org/10.1145/3364510.3364524

[8] Geela Fabic, Antonija Mitrovic, Kourosh Neshatian. Towards a Mobile Python Tutor: Understanding Differences in Strategies used by Novices and Experts. In: Proceedings of the 13th International Conference on Intelligent Tutoring Systems, LNCS, vol. 9684, pp. 447–448. Springer Heidelberg (2016)

[9] Rita Garcia, Katrina Falkner, and Rebecca Vivian. 2018. Scaffolding the Design Process using Parsons Problems. In Proceedings of the 18th Koli Calling International Conference on Computing Education Research (Koli Calling '18). Association for Computing Machinery, New York, NY, USA, Article 26, 1–2. DOI:https://doi.org/10.1145/3279720.3279746

[10] Juha Helminen, Petri Ihantola, Ville Karavirta, and Lauri Malmi. 2012. How do Students Solve Parsons Programming Problems?: An Analysis of Interaction Traces. In Proceedings of the ninth annual international conference on International computing education research (ICER '12). ACM, New York, NY, USA, 119-126. DOI: https://doi.org/10.1145/2361276.2361300.

[11] Petri Ihantola and Ville Karavirta. 2011. Two-dimensional Parson's Puzzles: The Concept, Tools, and First Observations. Journal of Information Technology Education. 10 (2011), 119–132.

[12] Petri Ihantola and Ville Karavirta. 2010. Open Source Widget for Parson's Puzzles. In Proceedings of the fifteenth annual conference on Innovation and technology in computer science education (ITiCSE '10). ACM, New York, NY, USA, 302-302. DOI: https://doi.org/10.1145/1822090.1822178

[13] Amruth N. Kumar. 2019. Mnemonic Variable Names in Parsons Puzzles. In Proceedings of the ACM Conference on Global Computing Education (CompEd '19). ACM, New York, NY, USA, 120-126. DOI: https://doi.org/10.1145/3300115.3309509

[14] Amruth N. Kumar. 2019. Helping Students Solve Parsons Puzzles Better. In Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '19). ACM, New York, NY, USA, 65-70. DOI: https://doi.org/10.1145/3304221.3319735

[15] Amruth N. Kumar. 2018. Epplets: A Tool for Solving Parsons Puzzles. In Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18). ACM, New York, NY, USA, 527-532. DOI: https://doi.org/10.1145/3159450.3159576.

[16] Amruth N. Kumar. 2017. The Effect of Providing Motivational Support in Parsons Puzzle Tutors. In Proceedings of Artificial Intelligence in Education. (AI-ED 2017), Wuhan, China, June 2017, 528-531. DOI= https://doi.org/10.1007/978-3-319-61425-0_56

[17] Amruth N. Kumar. 2019. Representing and Evaluating Strategies for Solving Parsons Puzzles. In Proceedings of Intelligent Tutoring Systems (ITS 2019), Kingston, Jamaica. Springer LNCS 11528, 193-203

[18] Raymond Lister, Tony Clear, Simon, Dennis J Bouvier, Paul Carter, Anna Eckerdal, Jana Jacková, Mike Lopez, Robert McCartney, Phil Robbins, Otto Seppälä, and Errol Thompson. 2010. Naturally Occurring Data as Research Instrument: Analyzing Examination Responses to Study the Novice Programmer. ACM SIGCSE Bulletin 41, 4 (2010), 156–173

[19] Mike Lopez, Jacqueline Whalley, Phil Robbins, and Raymond Lister. 2008. Relationships Between Reading, Tracing and Writing Skills in Introductory Programming. In Proceedings of the Fourth International Workshop on Computing Education Research (ICER '08). ACM, New York, NY, USA, 101-112. DOI=http://dx.doi.org/10.1145/1404520.1404531.

[20] Briana B. Morrison, Lauren E. Margulieux, Barbara Ericson, and Mark Guzdial. 2016. Subgoals Help Students Solve Parsons Problems. In Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16). ACM, New York, NY, USA, 42-47. DOI: https://doi.org/10.1145/2839509.2844617.

[21] Dale Parsons and Patricia Haden. 2006. Parson's Programming Puzzles: A fun and Effective Learning Tool for First

Programming Courses. In Proceedings of the 8th Australasian Conference on Computing Education - Volume 52 (ACE '06), Denise Tolhurst and Samuel Mann (Eds.), Vol. 52. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 157-163.

[22] Susan Wiedenbeck, Vennila Ramalingam, Suseela Sarasamma, Cynthia L. Corritore. A Comparison of the Comprehension of Object-oriented and Procedural Programs by Novice Programmers. Interacting with Computers, 11 (3). January 1999, Pages 255–282, https://doi.org/10.1016/S0953-5438(98)00029-0

[23] Amruth N. Kumar, Do Students use Semantics When Solving Parsons Puzzes? – A Log-Based Investigation. Proceedings of Intelligent Tutoring Systems (ITS 2021). LNCS 12677. June 2021. 444-450.

[24] Salil Maharjan and Amruth N. Kumar. Using Edit Distance Trails to Analyze Path Solutions of Parsons Puzzles", Proceedings of the 13th International Conference on Educational Data Mining (EDM 2020). July 2020, 638-642.

[25] Amruth N. Kumar. Using Markov Transition Matrix to Analyze Parsons Puzzle Solutions. Proceedings of the Educational Data Mining (EDM 2021) Workshop on Process Analysis Methods for Educational Data, Online, June 2021.

[26] Michael Kölling, Neil C. C. Brown, and Amjad Altadmri. 2015. Frame-Based Editing: Easing the Transition from Blocks to Text-Based Programming. In Proceedings of the Workshop in Primary and Secondary Computing Education (WiPSCE '15). ACM, New York, NY, USA, 29-38. DOI: https://doi.org/10.1145/2818314.2818331

[27] Valerie Barr and Deborah Trytten. 2016. Using Turing's craft Codelab to support CS1 students as they learn to program. ACM Inroads 7, 2 (May 2016), 67–75

[28] Robert S. Rist. (1989). Schema creation in programming. Cognitive Science, 13(3), 389-414.

[29] Chris Piech, Mehran Sahami, Daphne Koller, Steve Cooper, and Paulo Blikstein. 2012. Modeling how students learn to program. In Proceedings of the 43rd ACM technical symposium on Computer Science Education (SIGCSE '12). ACM, New York, NY, USA, 153-160.

[30] Robert S. Rist. (1991). Knowledge creation and retrieval in program design: A comparison of novice and intermediate student programmers. Human-Computer Interaction, 6(1), 1-46.

[31] Bassok, M., Chase, V.M., and Martin, S.A. (1998). Adding apples and oranges: Alignment of semantic and formal knowledge. Cognitive Psychology, 35(2), 99-134.

[32] Bassok, M., and Olseth, K.L. (1995). Object-based representations: Transfer between cases of continuous and discrete models of change. Journal of Experimental Psychology: Learning, Memory, and Cognition, 21(6), 1522-1538.

[33] Martin, S.A., and Bassok, M. (2005). Effects of semantic cues on mathematical modeling: Evidence from word-problem solving and equation construction tasks. Memory and Cognition, 33(3), 471-478.

[34] Christopher Watson and Frederick W.B. Li. 2014. Failure rates in introductory programming revisited. In Proceedings of the 2014 conference on Innovation & technology in computer science education (ITiCSE '14). ACM, New York, NY, USA, 39-44. DOI=http://doi.acm.org/10.1145/2591708.2591749

[35] Jens Bennedsen and Michael E. Caspersen. 2007. Failure rates in introductory programming. SIGCSE Bull. 39, 2 (June 2007), 32-36. DOI: https://doi.org/10.1145/1272848.1272879

Towards Scalable Adaptive Learning with Graph Neural Networks and Reinforcement Learning

Jean Vassoyan
Université Paris-Saclay,
CNRS, ENS Paris-Saclay,
Centre Borelli
Gif-sur-Yvette, France
onepoint
Paris, France
jean.vassoyan@ens-paris-saclay.fr

Jill-Jênn Vie
Inria Saclay – SODA
Palaiseau, France
jill-jenn.vie@inria.fr

Pirmin Lemberger
onepoint
Paris, France
p.lemberger@groupeonepoint.com

ABSTRACT

Adaptive learning is an area of educational technology that consists in delivering personalized learning experiences to address the unique needs of each learner. An important subfield of adaptive learning is learning path personalization: it aims at designing systems that recommend sequences of educational activities to maximize students' learning outcomes. Many machine learning approaches have already demonstrated significant results in a variety of contexts related to learning path personalization. However, most of them were designed for very specific settings and are not very reusable. This is accentuated by the fact that they often rely on non-scalable models, which are unable to integrate new elements after being trained on a specific set of educational resources. In this paper, we introduce a flexible and scalable approach towards the problem of learning path personalization, which we formalize as a reinforcement learning problem. Our model is a sequential recommender system based on a graph neural network, which we evaluate on a population of simulated learners. Our results demonstrate that it can learn to make good recommendations in the small-data regime.

Keywords

adaptive learning, learning path personalization, graph neural networks, reinforcement learning, recommender system

1. INTRODUCTION

Adaptive learning is an area of educational technology that focuses on addressing the unique needs, abilities, and interests of each individual student. This field emerged in the 1980s with the introduction of the first *Intelligent Tutoring Systems* (ITS) and experienced major expansion in the 1990s. As described by T. Murray in [20], an ITS usually consists of four components: a *domain model*, a *student model*, an *instructional model* and a *user interface model*. As we address the problem from an algorithmic point of view, we only focus on the first three models. The domain model is a representation of the knowledge to be taught; it often serves as a basis for the student model. The student model provides a characterization of each learner that allows to assess their knowledge and skills and anticipate their behavior. The instructional model takes the domain and student models as input to select strategies that will help each user achieve their learning objectives. This general structure allows ITSs to achieve many purposes (recommending exercises, providing feedback, facilitating memorization, etc.) while optimizing a variety of metrics (learning gains, engagement, speed of learning, etc.).

In this paper, we address the problem of learning path personalization with optimization of learning gains. This means that we look for a sequential recommender system that can provide each student with the right content at the right time (according to their past activity), in order to maximize their overall learning gains.

Towards this goal, "standard" approaches often require significant structuring of the domain model. This step is usually assisted by experts: they may be mobilized to tag educational resources, set up prerequisite relationships, draw up skill tables, etc. One example of such structuring is the Q-matrix [2] which maps knowledge components (KC) to exercises. These expert-based approaches present some serious practical limitations. First, they make it quite cumbersome to create resource sets, since each resource has to be properly tagged (sometimes with an extensive set of metadata). They also lead to poorly reusable recommender systems, since prerequisite relationships and skills maps are usually tailored to specific resource sets. This low reusability problem is often exacerbated by the modeling of resources/skills/KC as one-hot encodings [3, 21] which tie the model to a maximum number of resources/skills/KC it can handle. As a result, these approaches produce models that are not suitable for transfer learning. Our approach, on the other hand, is based on a graph neural network, which structure makes it possible to process data in a much more flexible way.

J. Vassoyan, J.-J. Vie, and P. Lemberger. Towards scalable adaptive learning with graph neural networks and reinforcement learning. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 351–361, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115663

Our contributions in this paper are threefold. First, we introduce a new setting for learning path personalization and formalize it as a model-free reinforcement learning (RL) problem. Second, we present a novel RL policy that can leverage educational resource content and users' feedback to make recommendations that improve learning gains. The proposed model has the advantage of being inherently scalable, reusable, and independent of any expert tagging. Third we evaluate our model on 6 semi-synthetic environments composed of real-world educational resources and simulated learners. The results demonstrate that it can learn to make good recommendations from few interactions with learners, thereby significantly outperforming the uniform random policy.

The rest of the paper is organized as follows. In Section 2, we relate our paper to prior research. In Section 3, we describe our setting, the assumptions we make and the problem we attempt to solve, which we formalize as a reinforcement learning problem. In Section 4, we present our novel RL policy. In Section 5, we describe our experimental setting and discuss our results. In Section 6 we address some limitations of our model and propose a few directions for future work. We finally conclude in Section 7.

2. RELATED WORK

In recent years, several works have used reinforcement learning to address the problem of learning path personalization. Most of these RL approaches are model-based, as they rely on a predefined student model to simulate student trajectories. However, no student model is completely accurate, and the learned instructional policies may overfit to the student model. Doroudi et al. [7] have attempted to learn policies that provide a better reward no matter the student model chosen (i.e. robust policies). Azhar et al. [1] proposed a method to gradually refine the student model by adding features that maximize the reward.

Reward functions usually involve learning gains. Subramanian and Mostow [29] defined learning gains as average difference between posterior and prior latent knowledge. Lan and Baraniuk [15] proposed to learn a policy for selecting learning actions so that the grade on the next exam is maximized. Clement et al. [5] attempted to optimize an increase in success rate in recent time steps, they used an ε-greedy approach. Doroudi et al. [8] conducted a thorough review of the different reward functions used in instructional policies.

The closest to our setting is probably the approach proposed by Bassen et al. [3] which, like ours, does not rely on expert pre-labeling of educational resources. However, in the absence of compensation for this lack of information, their reinforcement learning algorithm requires a substantial number of learners to converge to an effective policy: about 1000 learners for a corpus of 12 educational resources. Moreover, in their framework, educational activities were represented as one-hot encodings and passed to the policy via a fixed-size vector. Therefore, this approach does not allow to work with an evolving corpus of educational resources (which is the case for most *e-learning* platforms) nor to reuse the model on another set, unless it is completely re-trained.

In contrast, our approach leverages information from resource keywords which allows to achieve convergence in a relatively small number of episodes, while maintaining a high level of flexibility. This keyword-based approach was inspired by the work of Gasparetti et al. [12, 11]. Although the authors did not directly address the problem of learning path personalization, they outlined a method of feature extraction from textual resources that proved to be very successful in predicting prerequisite relationships.

3. PROBLEM FORMULATION

3.1 Description of the setting

Consider an *e-learning* platform with a collection of educational resources which have been designed to cover a specific topic, for example "an introduction to machine learning". Consider a population $\mathscr{P}$ of learners to be trained on this topic. The goal of learning path personalization is to be able to recommend a sequence of educational resources to each learner so as to maximize his overall *learning gains*. Therefore the resulting machine learning problem can be expressed in the following terms: given a large enough sample U of users from $\mathscr{P}$, how can we train a machine learning model to make recommendations to users from U so as to generalize to the whole population?

In this paper, we work at the scale of short learning paths ($\sim$ 1 hour), which means that each learning session only consists of a few interactions between the learner and the ITS. One advantage of this setting is that it reduces the effects of memory loss: we assume that when a learner visits a new resource, what he learned from the previous ones is still in his working memory.

We first make a few assumptions about the learning sessions:

(a_1) Each learner follows one learning path of equal length (i.e. same number of resources). The purpose of this assumption is primarily to simplify the notations as it can be easily relaxed without making major modifications to the model.

(a_2) There is no interaction between the learner and the external world (no communication, no access to external resources). This makes it possible to work in the closed system {learner + ITS}. While incorrect in most cases, this assumption may be more reasonable in our setting than in a multi-day learning context.

(a_3) We assume the existence of a feedback signal that provides information about user understanding of each resource. This signal can take three values:

- ($f_<$): the user did not understand the resource
- ($f_>$): the user understood, but found it too easy
- ($f_\circ$): the resource was at the right level.

In practice, such feedback can be obtained from self-assessment or more sophisticated test, and should be associated with an error margin to account for its imprecision. Nevertheless, in this study, we assume that each feedback is perfectly accurate.

A view of such a learning session is provided in Figure 1.

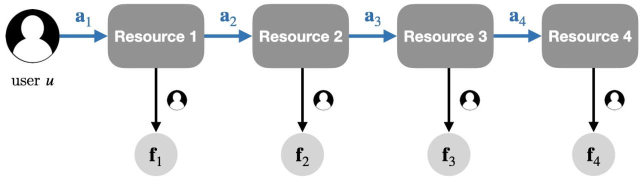


Figure 1: A view of a learning session. In this example, the session length is $T = 4$. Actions $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4$ are the recommendations of the ITS. $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4$ are the feedback signals returned by the user.

To further simplify the problem, we also adopt a few simplifying assumptions about the educational resources:

(a_4) They are purely textual resources, written in natural language. We indeed consider that most educational formats can be easily transcribed into text (transcript of a video, legend of a diagram, caption of an image etc.).

(a_5) They are *self-contained*, which means that they can be considered independently. This implies for example that they do not explicitly refer to each other. Although quite strong, this assumption is essential to prevent mandatory dependencies and foster diversity of learning paths.

(a_6) Each resource explains one or few concepts and has equivalent "educational value". This involves that each resource carries the same "amount" of knowledge.

Some examples of educational resources that satisfy these requirements are provided in Figure 4 of the Appendix.

Our goal with this work is to design a machine learning algorithm that can leverage learners' feedback to text-based educational resources to model their understanding of each concept, anticipate their reactions, and recommend resources that maximize their overall learning gains.

Since most *e-learning* platforms are in constant evolution, our goal is not only to solve this problem but to do it in a flexible and scalable way. This means that the model should not require full retraining when new resources are added to (or removed from) the platform. Actually, it should be able to extrapolate to new resources what it learned from previous interactions. This suggests that the number of parameters of our model should not depend on the size of the corpus.

3.2 Formalization

In this section, we formalize the problem described above as a reinforcement learning problem. We use the terms "user" and "learner" interchangeably to refer to any individual from the sample U. Similarly, we refer to an educational resource with the terms "document" or "resource".

In the following, we denote: T the length of each learning session (identical for each user), $\mathscr{D}$ the corpus of documents, d a document from $\mathscr{D}$, u a user (or learner) from the sample U, $\mathbf{f}_d$ the feedback given by a learner on document d.

The sequential recommendation problem defined above can be easily expressed as a reinforcement learning problem where: the agent is the recommender system, the environment is the population $\mathscr{P}$ of students and each episode is a learning path. This problem can be formulated as a partially observable Markov decision process ($\mathcal{S}$, $\mathcal{A}$, $\mathcal{O}$, $\mathcal{T}$, $\mathcal{R}$, $\mathcal{Z}$) where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{O}$ is the observation space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ defines the conditional transition probabilities, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function and $\mathcal{Z} : \mathcal{S} \times \mathcal{A} \times \mathcal{O} \to [0, 1]$ is the observation function. More precisely, in our setting:

- $\mathbf{s}_t \in \mathcal{S}$ is the (unknown) knowledge state of the learner at step t;
- $\mathbf{a}_t \in \mathcal{A}$ is the document selected by the recommender system at step t; we can write $\mathbf{a}_t = \mathbf{d}_t$;
- $\mathbf{o}_t \in \mathcal{O}$ is the observation made at step t, which is a tuple of the selected document and the returned feedback: $\mathbf{o}_t = (\mathbf{d}_t, \mathbf{f}_t)$;
- $\mathcal{T}(\mathbf{s}, \mathbf{a}, \mathbf{s}') = \mathbb{P}(\mathbf{s}_{t+1} = \mathbf{s}' \mid \mathbf{s}_t = \mathbf{s}, \mathbf{a}_t = \mathbf{a})$ is unknown, as it represents the impact of selecting document $\mathbf{a}$ on learner's state $\mathbf{s}_t$;
- $\mathcal{Z}(\mathbf{s}, \mathbf{a}, \mathbf{o}) = \mathbb{P}(\mathbf{o}_{t+1} = \mathbf{o} \mid \mathbf{s}_{t+1} = \mathbf{s}, \mathbf{a}_t = \mathbf{a})$ is also unknown and represents the probability of observing $\mathbf{o}$ in state $\mathbf{s}$ after choosing document $\mathbf{a}$;
- $\mathcal{R}(\mathbf{s}_t, \mathbf{a}_t)$ is the learning gain of the user at step t, which we define as follows:
$$\mathcal{R}(\mathbf{s}_t, \mathbf{a}_t) = \mathbb{1}_{\{\mathbf{f}_t = f_\circ\}}. \tag{1}$$
We indeed consider that only feedback $f_\circ$ corresponds to an effective learning gain. We denote $\mathcal{R}(\mathbf{s}_t, \mathbf{a}_t) = \mathbf{r}_t$ in the following.

To solve this problem, we need to find a policy $\pi : \mathcal{O} \to \mathcal{A}$ that maximizes the expected return over each episode η:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\eta \sim \pi} \left[\sum_{t=1}^{T} \mathbf{r}_t \right]. \tag{2}$$

4. OUR RL MODEL

A common approach to solve partially observable Markov decision processes (POMDP) is to leverage information from past observations $\mathbf{o}_1, \ldots, \mathbf{o}_t$ to build an estimation of $\mathbf{s}_t$ which is then used to select the next action (illustrated in Figure 2). This boils down to encoding these observations into a latent space $\mathcal{S}$. In our setting, this latent space contains all possible knowledge states for the learner, which is why we call it *knowledge space* in the following.

4.1 Knowledge space

While more compact than the observation space, the knowledge space should be informative enough to convey a relevant approximation of learner's knowledge.

We decided to structure this representation with the keywords of the corpus, denoted $(w_1, \ldots, w_M)$. We define a

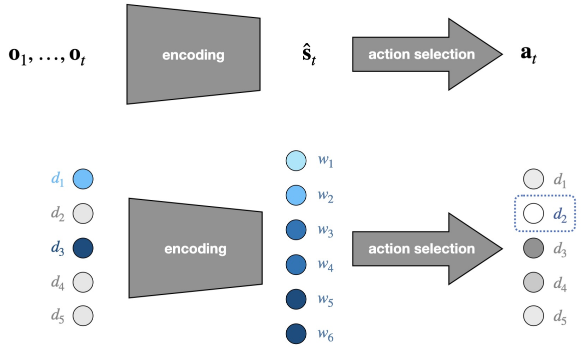


Figure 2: Up, a view of common policy architecture to solve POMDP. Down, this architecture applied to our setting.

keyword as a word or group of words that refers to a technical concept closely related to the subject of the corpus. Some examples of keywords extracted from educational resources are provided in the Appendix. The keywords carry information about the concepts addressed by the documents and are therefore a good approximation of their pedagogical content. That is why we modeled the knowledge state of each learner as a collection of vectors $(\mathbf{w}_1, \ldots, \mathbf{w}_M)$ which represents his "understanding" of each keyword. We indeed consider that a keyword can be understood in multiple ways depending on the context in which it occurs, and a multidimensional vector can be a convenient way to capture this plurality. This is illustrated in Figure 2. From this perspective, the knowledge space $\mathcal{S}$ can be defined as $\mathcal{S} := \underbrace{\mathbb{R}^K \times \cdots \times \mathbb{R}^K}_{M}$.

Note that we do not consider keyword extraction as a task requiring expert knowledge since it can be done by any creator of educational content and involves fewer skills than defining the knowledge components of a course. Moreover, it is mainly a pattern-matching task that can be automated through a keyword extraction algorithm [9, 22, 4].

4.2 Policy

Following the previous considerations, the policy π_θ should take a collection of observations $\mathbf{o}_1, \ldots, \mathbf{o}_t$ as input, encode it into the latent space $\mathcal{S}$ and return a recommendation for the next document $\mathbf{d}_t$. We emphasize that this function should also meet the aforementioned flexibility and scalability requirements.

A natural way to model the relationship between documents and keywords is to build a bipartite graph $\mathcal{G} = (\mathcal{V}_\mathcal{D}, \mathcal{V}_\mathcal{W}, \mathcal{E})$, where $\mathcal{V}_\mathcal{D}$ is the set of *document* nodes, $\mathcal{V}_\mathcal{W}$ is the set of *keyword* nodes and $\mathcal{E}$ is the set of edges, with $(v_d, v_w) \in \mathcal{E}$ if the document d contains the word w.

We chose to use a graph neural network (GNN) as a policy. GNNs are quite convenient for this task as they allow to enrich node features with information about their extensive neighborhood, through message-passing. Therefore, documents (respectively keywords) that share a large number of keywords (respectively documents) will also have similar embeddings. This allows to build keyword embeddings that contain information about feedback from neighboring documents $(\mathbf{o}_1, \ldots, \mathbf{o}_t \to \hat{\mathbf{s}}_t)$. Message-passing can also be used the other way around, from keywords to documents, to build embeddings that inform about the relevance of each document according to the estimated knowledge state $(\hat{\mathbf{s}}_t \to \mathbf{a}_t)$. Another significant advantage of GNNs is that their number of parameters does not depend on the size and structure of the graph, which makes them highly flexible and scalable.

Multiple options are possible for the initial node features. For keyword nodes, pre-trained word embeddings are a natural choice. As for the document nodes, a simple null vector is sufficient. However one may choose to include extra information about the documents if it is available (type of document, format, length etc.). We denote as $(\mathbf{x}_w)_{w \in \mathcal{V}_\mathcal{W}}$ and $(\mathbf{x}_d)_{d \in \mathcal{V}_\mathcal{D}}$ the initial feature vectors of keyword and document nodes.

In our model, we adapted a version of GAT (graph attention networks) [32] to the heterogeneity of our bipartite graph:

$$\forall d \in \mathcal{V}_\mathcal{D},\ \mathbf{h}_d^{(\ell+1)} = \sigma\left(\sum_{w \in \mathcal{N}(d)} \alpha_{dw}^{(\ell)} W_D^{(\ell)} \mathbf{h}_w^{(\ell)} + B_D^{(\ell)}\right) \quad (3)$$

$$\forall w \in \mathcal{V}_\mathcal{W}, \mathbf{h}_w^{(\ell+1)} = \sigma\left(\sum_{d \in \mathcal{N}(w)} \alpha_{wd}^{(\ell)} W_W^{(\ell)} \mathbf{h}_d^{(\ell+1)} + B_W^{(\ell)}\right) \quad (4)$$

$\mathbf{h}_d^{(\ell)} \in \mathbb{R}^K$ is the embedding of node d at ℓth layer, with $\mathbf{h}_d^{(0)} = \mathbf{x}_d$. $\mathcal{N}(d)$ is the set of neighbors of node d in the graph. $\alpha_{dw}^{(\ell)}$ is a *self-attention* coefficient, detailed in the Appendix. $\sigma(\cdot)$ is the ReLU activation function (rectified linear unit). $W_W^{(\ell)}$, $W_D^{(\ell)}$, $B_W^{(\ell)}$ and $B_D^{(\ell)}$ are trainable parameters. This back-and-forth mechanism between documents and keywords allows to learn distinct filters for each node type (document or keyword), effectively addressing the graph's heterogeneity. In the following, we refer to equations (3) and (4) as *bipartite GAT layers* and denote them (KW $\xrightarrow{(3)}$ DOC) and (DOC $\xrightarrow{(4)}$ KW). Note that they can be chained one after the other.

We define our first block of bipartite GAT layers as follows:

$$\text{BLOCK1} \ = \ \text{KW} \xrightarrow{(3)} \text{DOC} \xrightarrow{(4)} \text{KW} \xrightarrow{(3)} \text{DOC}. \quad (5)$$

After this block, document embeddings $(\mathbf{h}_d^{(2)})_{d \in \mathcal{V}_\mathcal{D}}$ contain information about keywords from their extended neighborhood. Using a Hadamard product, we enrich these embeddings with user feedback:

$$\mathbf{h}_d^{(\varphi)} = \mathbf{h}_d^{(2)} \odot \mathsf{MLP}_{K_d \to K}(\mathbf{f}_d) \quad (6)$$

$\mathbf{h}_d^{(2)}$ and $\mathbf{h}_d^{(\varphi)}$ are the embeddings of document d before and after adding the feedback. $\mathbf{f}_d$ is an encoding of user's feedback on document d, which is passed through a multilayer perceptron (MLP). We use a "not visited" feedback for the documents that the learner has not yet visited.

After doing this operation on each document node, we apply another block of bipartite GAT layers:

$$\text{BLOCK2} \ = \ \text{DOC} \xrightarrow{(4)} \text{KW} \xrightarrow{(3)} \text{DOC}. \quad (7)$$

Operation (4) allows to enrich keyword embeddings with feedback from neighboring documents, which carry informa-

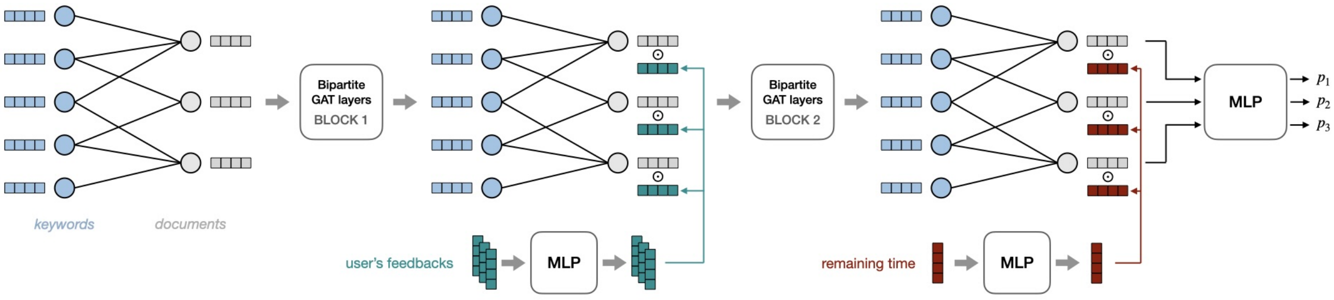


Figure 3: The architecture of our policy network on a 3-document corpus

tion about user's understanding. We consider these embeddings as a good approximation of learner's knowledge state, which is why we define $\hat{\mathbf{s}}_t := (\mathbf{h}_w^{(2)})_{w \in \mathcal{V}_\mathcal{W}}$. The final GAT layer (3) maps $\hat{\mathbf{s}}_t$ to documents for the next recommendation.

Before assigning probabilities to each document in the final step, we enrich document embeddings by incorporating information about the remaining time in the session, which, as we observed, slightly improved the performance of the model:

$$\mathbf{h}_d^{(\tau)} = \mathbf{h}_d^{(3)} \odot \mathsf{MLP}_{K_\tau \to K}(\Delta_t) \qquad (8)$$

$\mathbf{h}_d^{(3)}$ and $\mathbf{h}_d^{(\tau)}$ are the embeddings of document d before and after adding the remaining time. $\Delta_t = T - t$ is an encoding of the remaining time (or remaining steps) at step t.

Eventually, the embeddings $\mathbf{h}_d^{(\tau)}$ are passed through an MLP to assign a score to each document. These scores are converted into probabilities via a softmax over all document nodes (further details in the Appendix):

$$\pi_\theta\left(d \mid \mathbf{o}_1, \dots, \mathbf{o}_t\right) = \operatorname*{softmax}_{\mathcal{V}_\mathcal{D}} \left(\mathsf{MLP}_{K \to 1}(\mathbf{h}_d^{(\tau)})\right). \qquad (9)$$

The full architecture of the policy is illustrated in Figure 3.

4.3 RL Algorithm

As our policy selects the next action directly from observations, it belongs to the *policy-based* reinforcement learning paradigm, especially the *policy gradient* methods. The latter make it possible to maximize the expected return by optimizing directly the parameters of π_θ through gradient descent. We chose the `REINFORCE` algorithm [31] for its simplicity. At the end of each episode, π_θ is updated as follows:

$$\forall t \in [1, T], \quad \theta \leftarrow \theta + \lambda \nabla_\theta \log \pi_\theta\left(s_t, a_t\right) v_t \qquad (10)$$

with λ the learning rate and $v_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$ the return of the episode from step t.

Note that we could learn our policy using more sophisticated RL algorithms like actor-critic, which usually has lower variance. However, it is likely that the current architecture would provide a poor state value function as it only operates at the scale of node neighborhoods and does not have a "global" view of the graph. Some changes in this architecture might nevertheless be done to process information at a larger scale, as discussed in Section 6.

Table 1: Key statistics of each corpus

corpus	# doc	# kw	# edges	diameter
Corpus 1	33	68	154	10
Corpus 2	11	31	62	6
Corpus 3	19	39	83	8
Corpus 4	28	55	113	8
Corpus 5	18	41	66	∞
Corpus 6	20	45	143	6

5. EXPERIMENTS

Given the complexity of conducting mass experiments on real learners, we chose to evaluate our model in an environment made up of semi-synthetic data. Our implementation is written in Python and is available on GitHub[1]. We also provided the hyperparameters of our model in Table 4 of the Appendix.

5.1 Experimental setting

Linear corpus

We introduce what we call a "linear" corpus. Starting from a regular course divided into sections and subsections, we treat each subsection as one document. The corpus resulting from this decomposition is "linear", in the sense that it was designed to be followed in a single, pre-defined order, which is identical for each learner. Therefore, it leaves practically no room for personalization. Six corpora were constructed this way: three about data science (1-3) and three about programming (4-6). They were all built from courses taken from a popular *e-learning* platform.

For the purpose of our experiments, we have chosen to tag keywords "by hand" to avoid introducing any noise in the results. Our methodology was quite simple: for each document, we collected keywords referring to technical concepts related to the topic of the course. Table 1 presents some key statistics about each corpus and their associated bipartite graphs. Note that the graph of corpus 5 is disconnected: indeed, one of its documents only contains keywords that do not appear in any other document. Despite significantly complicating the task for a diffusion model like ours, we have chosen to keep this corpus for our experiments.

[1] https://github.com/jvasso/graph-rl4adaptive-learning

Simulated learners

Since each corpus has been designed to be explored in a single pre-defined order, we assume that the only way to understand it is to follow this order scrupulously. Therefore we have decided to simulate the behavior of learners in this very simple way: as long as the policy recommends documents in the right order, the learner returns the feedback ($f_\circ$). Conversely, each time the algorithm recommends a document too early or too late, the learner returns the feedback ($f_<$) or ($f_>$). A detailed example is given in the Appendix.

Since our simulated learners have a straightforward behavior, the purpose of this experiment is not to evaluate the personalization or generalization capabilities of our model, but to assess its ability to grasp the structure of a corpus, by finding its original order in a reasonable number of episodes (i.e. a few learners). While trivial at first glance, this task can be quite difficult for an RL agent in the small-data regime. Besides, each corpus contains some parts that are independent of each other which suggests that in practice, multiple learning trajectories might be understandable to real learners. From this perspective, the "strict" feedback of our simulated learners can distort the real nature of the relationships between resources and make the task more difficult for our recommender system.

Policy

In our experiments, we compared 3 different policies. The first one is the uniform random policy. The second one is our policy with one-hot-encodings as keyword features. The third one is our policy with Wikipedia2Vec embeddings [34] as keyword features. Wikipedia2Vec embeddings are quite suitable for our task as they contain encyclopedic information about the relationship between words and entities. They were derived from a skip-gram model trained on a triple objective, which is detailed in the Appendix. We used null vectors as document features for each policy.

Training

In each experiment, the maximum achievable return is equal to the size of the corpus. We set the horizon T to the size of the corpus to make sure that only an optimal policy (i.e. one that makes no "mistake") can reach this return. In this setting, the return of the random policy follows a binomial distribution with parameters (T, $\frac{1}{T}$). Therefore its expected return is 1 for each episode. We also set the discount factor $\gamma = 0$ during training because in this very specific setting, the best action at each step t can be learned from immediate reward. We trained our model from scratch over 50 episodes ($\sim$ 50 students) for each corpus, with a constant learning rate.

5.2 Results

Since the REINFORCE algorithm has quite a high variance, we averaged each episodic return over 25 random seeds. The resulting learning curves are shown in Figure 6 of the Appendix and the last episodic returns (measured at 50th episode) are reported in Table 2.

From these curves, one can notice that despite the small-data regime and the choice of a sub-optimal RL algorithm (the REINFORCE algorithm is known to be quite unstable and sample-inefficient), our agent succeeded in recovering a significant part of the original order of each corpus. Most of the time, it achieved average return over 10 whereas the random policy was stuck in an expected return of 1.

Best performance was achieved on Corpus 2. Indeed, it is the only one for which our model managed to reach the maximum achievable return most of the time. This may be partly due to the small number of documents in this corpus. However, we stress that the number of documents alone is not a sufficient feature to account for the variability of the results. For instance, corpora 3 and 6 have a nearly similar number of documents, but our model performed very differently on these two corpora. Moreover, in the case of the Wikipedia2Vec approach, it is not guaranteed that a large corpus should be more difficult than a small one, since the episodes are shorter for small corpora and therefore the algorithm has fewer steps to grasp the geometrical structures in the distribution of Wikipedia2Vec embeddings.

The diameter of the graph may also impact the performance of the model. Indeed, Corpus 2 is again the one with the smallest diameter, which may have helped the model to determine the relationships between documents and keywords more quickly. However, this must be balanced with the results on Corpus 6, on which our model performed far worse (in terms of normalized return) despite equal diameter.

Another noticeable result is the one of Corpus 5. We remind that this corpus was the only one to be disconnected. Actually, it was disconnected at the 11th document, which is consistent with the performance of the model: indeed, episodic return lower than 10 indicates that it failed to make recommendations beyond the 10th document. This can be explained quite simply: since this document is disconnected from the rest of the graph, it does not benefit from message-passing and therefore receives no information about other documents feedback.

Eventually, one cannot ignore the extremely high variance of the episodic return for almost all corpora (except for Corpus 2). This is partly due to the choice of the REINFORCE algorithm, which is known for its high instability.

Ablation study

We conducted an ablation study to analyse the contribution of Wikipedia2Vec embeddings compared to simple one-hot encodings. Even though the approach with embeddings

Table 2: Comparison between episodic returns when using Wikipedia2Vec and one-hot encodings as keyword features

Corpus	Wikipedia2Vec	One-hot encodings
Corpus 1	$\mathbf{16.48 \pm 2.66}$	13.36 ± 1.74
Corpus 2	$\mathbf{10.84 \pm 0.14}$	10.28 ± 0.37
Corpus 3	$\mathbf{14.40 \pm 1.31}$	11.68 ± 1.13
Corpus 4	$\mathbf{15.16 \pm 0.90}$	12.52 ± 0.98
Corpus 5	$\mathbf{9.80 \pm 2.13}$	7.56 ± 1.83
Corpus 6	$\mathbf{11.24 \pm 1.52}$	8.24 ± 0.84

performed significantly better on each corpus, the high error margins and the similarity between trends suggest that our model was not truly able to leverage high level information about the relationships between Wikipedia entities. Instead, it is more likely that it simply "overfit" to each corpus. This lack of generalization is not a problem in the setting of our experiment but can be a serious issue in transfer learning scenarios and therefore needs to be addressed.

6. LIMITATIONS AND FUTURE WORK

6.1 Size and structure of the graph

All of our experiments have been conducted on small graphs (less than ~ 100 nodes). However, it is likely that our model would struggle a little more on larger graphs as the receptive field of each node accounts for a smaller fraction of the graph in such case. Besides, it is not possible to increase the depth of a GNN indefinitely because of the over-smoothing problem [14, 33, 16]. Therefore, it is likely that these embeddings alone would not be sufficiently informative to allow for long-term planning. This limitation can be addressed with down- and upsampling methods such as pooling and unpooling operations on graphs, which make it possible to process information at multiple scales [6, 28, 35]. It can also be addressed with planning techniques such as Monte Carlo Tree Search, which has demonstrated great performance in combination with deep RL techniques [23, 26, 27].

As we saw in subsection 5.2, there is also an issue with disconnected graphs since our model failed to make predictions beyond the disconnected document node in Corpus 5. One possible solution could be to slightly modify the structure of the graph, for example through link prediction based on keyword embeddings.

Eventually, it is important to note that we tested our approach on corpora related to engineering topics — machine learning and programming — which keyword distributions might be quite similar (cf. Figure 5 in the Appendix). Yet, corpora related to different topics may have completely different keyword distributions. Therefore, it would be worth comparing the performance of the model on a wider range of subjects in the future.

6.2 Variance and sample efficiency

As stated in Section 5, our approach suffers from high variance, partly due to the choice of the `REINFORCE` algorithm. Some other on-policy methods have demonstrated great success in reducing variance [24, 25, 18]. Nevertheless, these approaches remain generally not very sample-efficient. To improve sample-efficiency, it is quite common to use off-policy algorithms as they allow to reuse past experience [19, 17, 13]. However, as stated in Section 4, the implementation of an approximate Q-value function with a GNN is not trivial as it requires to leverage information at the scale of the entire graph, which involves modifications in the model. Another alternative is to use a model-based reinforcement learning algorithm (MBRL) [30, 23]. As they allow to learn a model of the environment (i.e. a model that predicts the next observations and rewards), MBRL techniques enable to reuse past experience and learn from a richer signal than the reward signal alone. Therefore, they are usually much more sample-efficient than model-free RL techniques. These approaches might be more appropriate in our case, as a local model like a GNN may more easily predict immediate feedback than the (long-term) *value* of a state-action pair.

6.3 Interpretability

One of the main limitations of our approach is its lack of interpretability. Ideally, an ITS would not only provide a personalized learning experience but also inform the learner about their progress and level of understanding, in order to encourage self-awareness and self-regulation. This is usually done with an *open learner model*. However, like most deep learning approaches, our recommender system is a black-box model and does not allow for easy interpretation. Yet, we hypothesize that the estimated knowledge state $\hat{\mathbf{s}}_t$ does not only contain semantic information about keywords but also about the way they were understood by the learner. Therefore, future work may consist in projecting these keyword embeddings into lower dimensional space to visualize their evolution throughout learning sessions.

6.4 Reusability

We designed a model that is flexible enough to be *theoretically* capable of transferring its knowledge from one corpus to another. However, this is only possible if the model has managed to capture high-level information that is common to all corpora. Unfortunately, our experiments do not allow to truly evaluate the transfer learning capabilities of our model. However, since it seems to overfit to the structure of each corpus, it might not have learned that much about the high-level relationships in the distribution of Wikipedia2Vec embeddings. Therefore, transfer learning might not be very effective in this case. Future directions to reduce overfitting may consist in applying regularization techniques to GNN (such as node dropout), or using training techniques that push the model to learn higher-level knowledge, such as meta-learning for RL [10].

7. CONCLUSION

In this paper, we presented a new model for learning path personalization, designed to be reusable and independent of any expert labeling. We demonstrated its ability to learn to make recommendations in 6 semi-synthetic environments made-up of real-world educational resources and simulated learners. Since this model is theoretically capable of transferring its knowledge from one corpus to another, it is a first step towards an approach that could considerably reduce the cold-start problem. Future work will investigate its performance in the context of transfer learning and with real students.

8. ACKNOWLEDGMENTS

We would like to warmly thank Vincent François-Lavet from VU Amsterdam who gave us some great advice on the formalization of the reinforcement learning problem. We would also like to thank Nicolas Vayatis, Argyris Kalogeratos (Centre Borelli), Nathanaël Beau and Antoine Saillenfest (onepoint) for their insightful reviews of the paper.

9. REFERENCES

[1] A. Z. Azhar, A. Segal, and K. Gal. Optimizing representations and policies for question sequencing using reinforcement learning. In A. Mitrovic and

N. Bosch, editors, *Proceedings of the 15th International Conference on Educational Data Mining*, pages 39–49, Durham, United Kingdom, July 2022. International Educational Data Mining Society.

[2] T. Barnes. The q-matrix method: Mining student response data for knowledge. In *American Association for Artificial Intelligence 2005 Educational Data Mining Workshop*, pages 39–46. AAAI Press, Pittsburgh, PA, USA, 2005.

[3] J. Bassen, B. Balaji, M. Schaarschmidt, C. Thille, J. Painter, D. Zimmaro, A. Games, E. Fast, and J. C. Mitchell. Reinforcement learning for the adaptive scheduling of educational activities. In R. Bernhaupt, F. F. Mueller, D. Verweij, J. Andres, J. McGrenere, A. Cockburn, I. Avellino, A. Goguey, P. Bjøn, S. Zhao, B. P. Samson, and R. Kocielnik, editors, *CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020*, pages 1–12. ACM, 2020.

[4] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt. YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289, 2020.

[5] B. Clément, D. Roy, P.-Y. Oudeyer, and M. Lopes. Multi-armed bandits for intelligent tutoring systems. *Journal of Educational Data Mining*, 7(2):20–48, 2015.

[6] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3837–3845, 2016.

[7] S. Doroudi, V. Aleven, and E. Brunskill. Robust evaluation matrix: Towards a more principled offline exploration of instructional policies. In C. Urrea, J. Reich, and C. Thille, editors, *Proceedings of the Fourth ACM Conference on Learning @ Scale, L@S 2017, Cambridge, MA, USA, April 20-21, 2017*, pages 3–12. ACM, 2017.

[8] S. Doroudi, V. Aleven, and E. Brunskill. Where's the reward? A Review of Reinforcement Learning for Instructional Sequencing. *International Journal of Artificial Intelligence in Education*, 29(4):568–620, 2019.

[9] P. Ferragina and U. Scaiella. TAGME: on-the-fly annotation of short text fragments (by Wikipedia entities). In J. Huang, N. Koudas, G. J. F. Jones, X. Wu, K. Collins-Thompson, and A. An, editors, *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 1625–1628. ACM, 2010.

[10] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 2017.

[11] F. Gasparetti, C. De Medio, C. Limongelli, F. Sciarrone, and M. Temperini. Prerequisites between learning objects: Automatic extraction based on a machine learning approach. *Telematics and Informatics*, 35(3):595–610, 2018.

[12] F. Gasparetti, C. Limongelli, and F. Sciarrone. Exploiting Wikipedia for discovering prerequisite relationships among learning objects. In *2015 International Conference on Information Technology Based Higher Education and Training, ITHET 2015, Lisbon, Portugal, June 11-13, 2015*, pages 1–6. IEEE, 2015.

[13] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1856–1865. PMLR, 2018.

[14] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings.* OpenReview.net, 2017.

[15] A. S. Lan and R. G. Baraniuk. A contextual bandits framework for personalized learning action selection. In T. Barnes, M. Chi, and M. Feng, editors, *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016, Raleigh, North Carolina, USA, June 29 - July 2, 2016*, pages 424–429. International Educational Data Mining Society (IEDMS), 2016.

[16] Q. Li, Z. Han, and X. Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In S. A. McIlraith and K. Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3538–3545. AAAI Press, 2018.

[17] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. In Y. Bengio and Y. LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[18] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In M. Balcan and K. Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1928–1937. JMLR.org, 2016.

[19] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level

control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[20] T. Murray. Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education (IJAIED)*, 10:98–129, 1999.

[21] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 505–513, 2015.

[22] S. Rose, D. Engel, N. Cramer, and W. Cowley. Automatic keyword extraction from individual documents. In *Text mining: applications and theory*, pages 1–20. Wiley Online Library, 2010.

[23] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, et al. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[24] J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, and P. Moritz. Trust region policy optimization. In F. R. Bach and D. M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1889–1897. JMLR.org, 2015.

[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. arXiv, abs/1707.06347, 2017.

[26] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[27] D. Silver and J. Veness. Monte-Carlo Planning in Large POMDPs. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 2164–2172. Curran Associates, Inc., 2010.

[28] M. Simonovsky and N. Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 29–38. IEEE Computer Society, 2017.

[29] J. Subramanian and J. Mostow. Deep reinforcement learning to simulate, train, and evaluate instructional sequencing policies. Spotlight presentation at Reinforcement Learning for Education workshop at Educational Data Mining 2021 conference, 2021.

[30] R. S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*, pages 216–224. Elsevier, 1990.

[31] R. S. Sutton, D. A. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In S. A. Solla, T. K. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems 12, NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999*, pages 1057–1063. The MIT Press, 1999.

[32] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

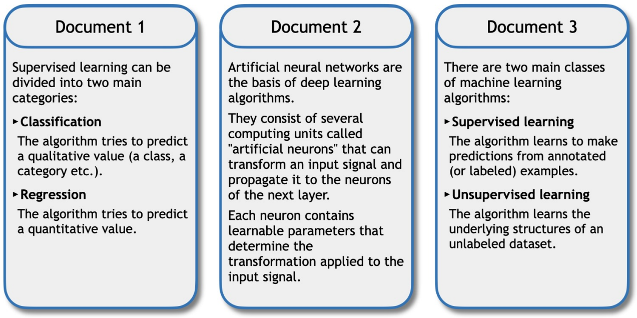
[33] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[34] I. Yamada, A. Asai, J. Sakuma, H. Shindo, H. Takeda, Y. Takefuji, and Y. Matsumoto. Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 23–30, Online, 2020. Association for Computational Linguistics.

[35] Z. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec. Hierarchical graph representation learning with differentiable pooling. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 4805–4815, 2018.

APPENDIX

Corpus and keywords. Some examples of educational resources that satisfy the assumptions (a_4), (a_5) and (a_6) described in Section 3.1 are provided in Figure 4. In the document 1, an appropriate collection of keywords would be: $\{$*supervised learning*, *classification*, *regression*$\}$.



Document 1

Supervised learning can be divided into two main categories:

- **Classification**

  The algorithm tries to predict a qualitative value (a class, a category etc.).

- **Regression**

  The algorithm tries to predict a quantitative value.

Document 2

Artificial neural networks are the basis of deep learning algorithms.

They consist of several computing units called "artificial neurons" that can transform an input signal and propagate it to the neurons of the next layer.

Each neuron contains learnable parameters that determine the transformation applied to the input signal.

Document 3

There are two main classes of machine learning algorithms:

- **Supervised learning**

  The algorithm learns to make predictions from annotated (or labeled) examples.

- **Unsupervised learning**

  The algorithm learns the underlying structures of an unlabeled dataset.

Figure 4: Three examples of *self-contained* educational resources taken from a corpus dealing with machine learning basics

Linear corpus. In our experiments, we used 6 corpora based on courses taken from a popular *e-learning* platform. Figure 5 shows the evolution of the total number of keywords throughout each course. Note that although they all cover different topics and were designed by different educators, they always introduce new keywords in a "linear" way. This supports the idea that the distribution of keywords can be a good indicator of pre-requisite relationships between documents.

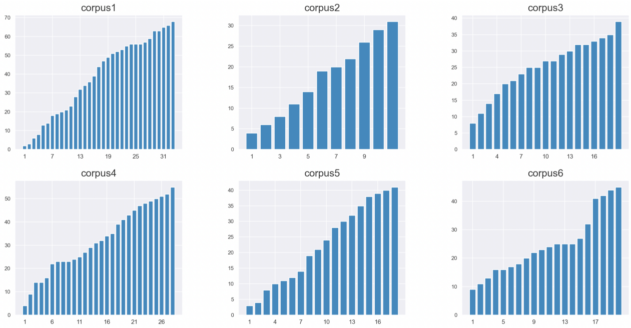


Figure 5: Evolution of the total number keywords in each course

Simulated learners. In the following we present a step by step example of a learning path followed by a simulated learner (also detailed in Table 3).

Table 3: Example of a sequence of interactions (learning path) between a simulated student and our policy

step	action $\mathbf{a}_t$	feedback $\mathbf{f}_t$	reward $\mathbf{r}_t$	$\mathcal{D}_\circ$
1	d_2	$f_<$	0	$\{\}$
2	d_1	$f_\circ$	1	$\{d_1\}$
3	d_3	$f_<$	0	$\{d_1\}$
4	d_2	$f_\circ$	1	$\{d_1, d_2\}$
5	d_1	$f_>$	0	$\{d_1, d_2\}$
6	d_3	$f_\circ$	1	$\{d_1, d_2, d_3\}$

Consider a corpus of three documents $\{d_1, d_2, d_3\}$, designed to be explored in the order of indices: d_1 is a prerequisite for d_2 and d_2 is a prerequisite for d_3. A simulated student can understand a document only if they have understood its prerequisites. Throughout the learning path, we maintain a set $\mathcal{D}_\circ$ of understood documents, initialized as an empty set: $\mathcal{D}_\circ = \{\}$.

At step 1, the policy recommends document d_2 (with prerequisite d_1). $d_1 \notin \mathcal{D}_\circ$, therefore the student returns feedback $(f_<)$. At step 2, the policy recommends document d_1. This document has no prerequisite, therefore the student returns feedback $(f_\circ)$ and we add d_1 to $\mathcal{D}_\circ$. At step 3, the policy recommends document d_3. $d_2 \notin \mathcal{D}_\circ$, therefore the student returns feedback $(f_<)$. At step 4, the policy recommends document d_2. $d_1 \in \mathcal{D}_\circ$, therefore the student returns feedback $(f_\circ)$ and we add d_2 to $\mathcal{D}_\circ$. At step 5, the policy recommends document d_1. $d_1 \in \mathcal{D}_\circ$, therefore the student returns feedback $(f_>)$. Finally at step 6, the policy recommends document d_3. $d_2 \in \mathcal{D}_\circ$, therefore the student returns feedback $(f_\circ)$ and we add d_3 to $\mathcal{D}_\circ$.

Note that in this example, we fixed $T = 6$ to display a greater number of situations. Conversely, in our experiments, T was always equal to the size of the corpus.

Self-attention. The self-attention coefficient α_{wd} used in Equations (3) and (4) is defined as follows. For any nodes w, d:

$$\alpha_{wd}^{(\ell)} = \operatorname*{softmax}_{\mathcal{N}(w)} \left(a \left(W^{(\ell)} \mathbf{h}_d^{(\ell)}, W^{(\ell)} \mathbf{h}_w^{(\ell)} \right) \right) \tag{11}$$

where $W^{(\ell)} \in \mathbb{R}^{K \times K}$ refers to the weights of ℓth layer and $a : \mathbb{R}^K \times \mathbb{R}^K \to \mathbb{R}$ is the additive attention mechanism. The softmax function is taken over all neighbors of node w (further details below).

Multilayer perceptron. Each $\mathsf{MLP}_{K_1 \to K_2}(\cdot)$ operator used in Section 4 is a multilayer perceptron with one hidden layer. For any input vector $\mathbf{x}$, this operation boils down to:

$$\mathbf{x}' = A^{(2)} \sigma \left(A^{(1)} \mathbf{x} + B^{(1)} \right) + B^{(2)} \tag{12}$$

where $\sigma(\cdot)$ is the ReLU activation function, $A^{(1)} \in \mathbb{R}^{K_1 \times K}$, $A^{(2)} \in \mathbb{R}^{K \times K_2}$, $B^{(1)} \in \mathbb{R}^K$ and $B^{(2)} \in \mathbb{R}^{K_2}$ are trainable parameters.

Softmax operator. The softmax operator over a finite collection E of real numbers is defined as follows:

$$\forall x \in E, \quad \operatorname*{softmax}_E(x) = \frac{\exp x}{\sum_{y \in E} \exp y}. \tag{13}$$

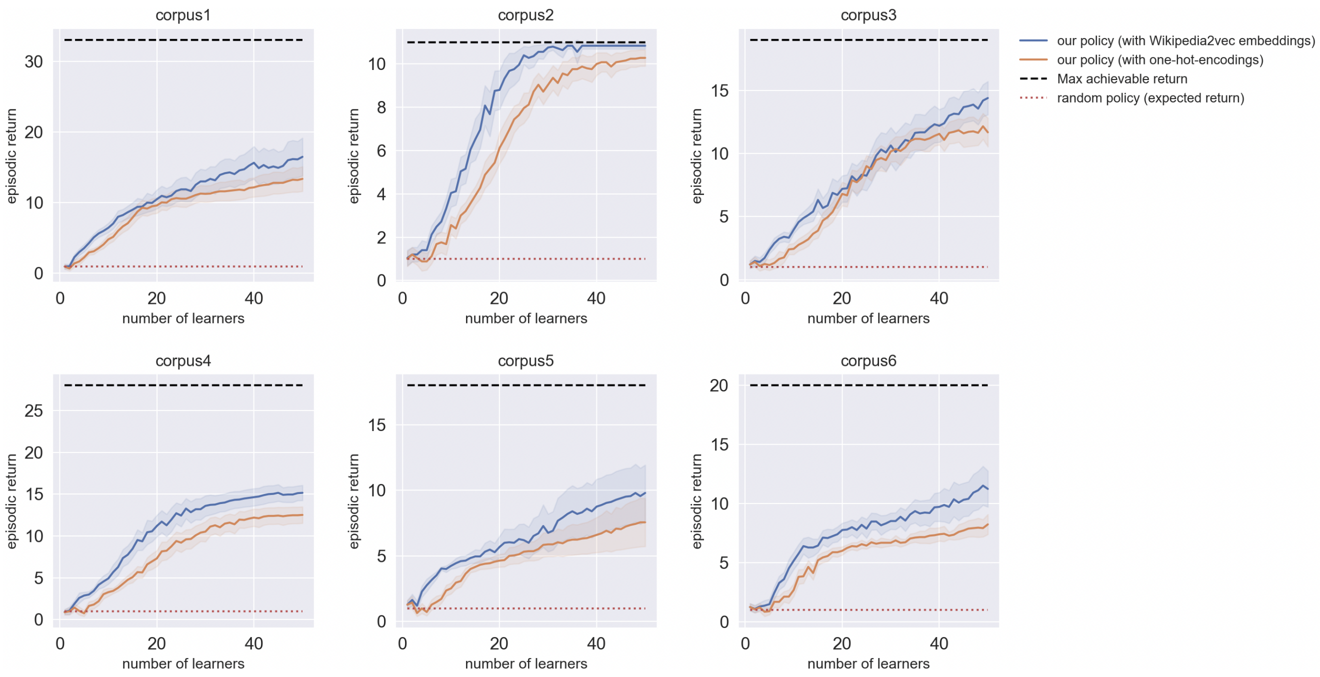


Figure 6: Evolution of the episodic return on 50 simulated learners for 6 corpora

Wikipedia2Vec. The pretrained Wikipedia2Vec embeddings leveraged as keyword features in our experiment were derived from a skip-gram model trained on a triple objective: (1) predicting neighboring entities in the link graph of Wikipedia, (2) predicting neighboring words given each word in a text contained on a Wikipedia page, and (3) predicting neighboring words given a target entity using anchors and their context words in Wikipedia [34]. We hypothesize that in addition to modeling the semantic information carried by each keyword, these embeddings allow to capture prerequisite relationships between concepts, especially through task (1).

Experimental results. The learning curves of our experiments are reported in Figure 6. For reproducibility, we also reported the hyperparameters of our model in Table 4.

Table 4: Hyperparameters used in our policy model

Name	Value
Learning rate	0.0005
Hidden dimension	32
Activation function	ReLU
Attention type	additive
Number of attention heads	2
Wikipedia2Vec embedding size	100
Documents encoding	vector of zero
Feedback encoding	one-hot-encoding
Remaining time encoding	counter
Batch size	16
Repeat per collect	15
Episodes per collect	1

Auto-scoring Student Responses with Images in Mathematics

Sami Baral
Worcester Polytechnic Institute
sbaral@wpi.edu

Anthony Botelho
University of Florida
abotelho@coe.ufl.edu

Abhishek Santhanam
Worcester Polytechnic Institute
asanthanam@wpi.edu

Ashish Gurung
Worcester Polytechnic Institute
agurung@wpi.edu

Li Cheng
Worcester Polytechnic Institute
lcheng1@wpi.edu

Neil Heffernan
Worcester Polytechnic Institute
nth@wpi.edu

ABSTRACT

Teachers often rely on the use of a range of open-ended problems to assess students' understanding of mathematical concepts. Beyond traditional conceptions of student open-ended work, commonly in the form of textual short-answer or essay responses, the use of figures, tables, number lines, graphs, and pictographs are other examples of open-ended work common in mathematics. While recent developments in areas of natural language processing and machine learning have led to automated methods to score student open-ended work, these methods have largely been limited to textual answers. Several computer-based learning systems allow students to take pictures of hand-written work and include such images within their answers to open-ended questions. With that, however, there are few-to-no existing solutions that support the auto-scoring of student hand-written or drawn answers to questions. In this work, we build upon an existing method for auto-scoring textual student answers and explore the use of OpenAI/CLIP, a deep learning embedding method designed to represent both images and text, as well as Optical Character Recognition (OCR) to improve model performance. We evaluate the performance of our method on a dataset of student open-responses that contains both text- and image-based responses, and find a reduction of model error in the presence of images when controlling for other answer-level features.

Keywords

Auto-scoring, Open-ended responses, Image responses, Online Learning Platform

1. INTRODUCTION

The blending of educational technologies with machine learning and statistical modeling has led to the emergence of tools designed to augment instruction. While some such tools are designed to automate certain tasks for the teacher (e.g. [3, 17, 2]), others attempt to improve the efficiency with which teachers are able to assess student work and write directed feedback to guide learning.

In the context of mathematics education, teachers utilize a range of question formats to assess students' understanding of covered topics. Prior work has described these question types in terms of "close-ended" and "open-ended" problems, distinguishing various types of problems by the difficulty with which answers to such questions may be automatically assessed by a simple matching algorithm. Multiple choice or fill-in-the-blank problems, as examples of close-ended problems, often allow for a small number of acceptable "correct" answers (i.e. in most cases, there is a single answer considered as correct). Although prior works have demonstrated the utility of these types of answers for measuring student knowledge (e.g. the extensive work on knowledge tracing [9, 25]), teachers often rely on the use of open-ended problems to gain deeper insights into the processes and strategies employed by students to solve such problems, as well as their ability to articulate their approach using proper mathematical terminologies. Short answer and essay question types are common in this regard, often with prompts such as "explain your reasoning", but other open-ended formats are also common in the domain of mathematics.

For mathematics, teachers often rely on the use of visual representations in conveying mathematical concepts. The use of diagrams, number lines, graphs, tables, and sometimes even pictographs are commonly used to portray numerical and algebraic relationships. Just as these are used for instruction, students are also commonly asked to generate these types of visual representations to demonstrate their understanding. While open-ended work has typically referred to the use of text and natural language within prior research (e.g. [13, 37, 4]), the definition extends to drawings and similar artifacts produced by students. Tools such as GeoGebra[18] and Desmos[12] are examples of computer-based applications that allow students to interact with graphs and algebraic expressions. While tools like these exist, many teachers still prefer to use more traditional technologies, often in the form of paper and pencil or other physical media (e.g. blocks) in conjunction with computer-based technologies; some systems encourage this blending of media by allowing students to take pictures of their work and upload them as responses to open-ended problems.

S. Baral, A. Botelho, A. Santhanam, A. Gurung, L. Cheng, and N. Heffernan. Auto-scoring student responses with images in mathematics. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 362–369, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115645

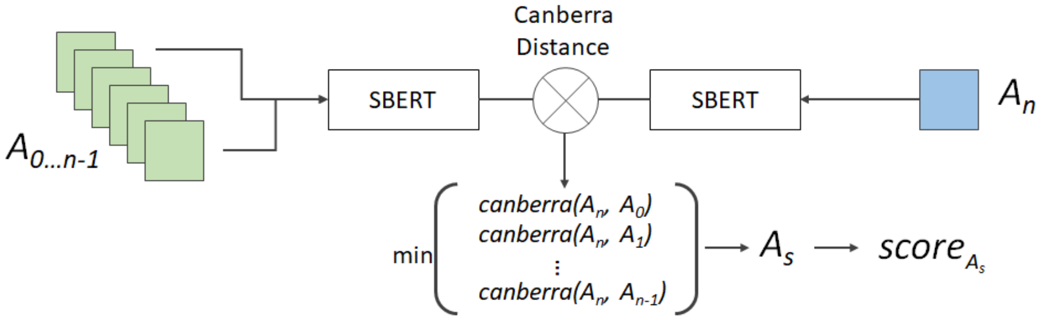


Figure 1: Simplified representation of the SBERT-Canberra method to generate a predicted score by identifying the most similar historic response to a given new student answer using Canberra distance within an embedding space.

This paper builds on prior work which focused on the development of an automated scoring tool for student answers to open response problems in mathematics [4]. Baral et. al, reported on how many students responded to open-response problems with images of their work (in the form of written mathematical equations and expressions as well as drawings of graphs, number lines, and other visual representations), whereas several others preferred to respond with a combination of an image of their work combined with a typed textual explanation within a single student response (e.g. the student draws a graph, uploads the image and then types a description of their thought process with the image of the graph). These cases were, unsurprisingly, found to contribute significantly to the model error as the presence of images in student responses were not previously accounted for within the developed methods. This work seeks to take initial steps toward understanding how recent advancements in areas of deep learning-based image and text embedding methods may help to address these challenges.

Specifically, this paper addresses the following research questions:

1. Does the use of pre-trained deep learning image and text embedding methods lead to improved performance in the context of previously-developed open response scoring models?
2. Are there differences in terms of the resulting model performance when comparing across different types of image-supporting embedding methods?
3. Does the incorporation of image-supporting embedding methods reduce the correlation between the presence of images in student responses and modeling error when accounting for other answer-level covariates?

2. RELATED WORKS

2.1 Automated Scoring Models

With the development of online learning platforms, there has been a growing body of research in the development of automated methods of assessment for analyzing and providing immediate feedback on students' work. These developments have prevailed in multiple domains of science [23, 6], programming[24, 26, 35], writing[21, 1, 8, 29, 39], mathematics[22, 13, 4] and college level courses[11]. In the domain of mathematics, auto-scoring have been developed for closed-ended problems with single or limited correct answers(e.g., multiple-choice question, fill-in-the-blank, check all that apply) [3, 17] to more open-ended problems with multiple possible solutions (eg. short answer, long answer, Explain in plain english.) [22, 13, 14, 4, 37, 38, 5, 32]. Some of these works support pure mathematical content [22], while others support combination of both mathematical and textual answers[13, 4, 5, 38]. However, most of these auto-scoring methods in mathematical domains are limited to either text or mathematical content, and a very few have started focusing on automating responses for image-based responses.

2.2 Methods for Image Analysis and Representation

Optical Character Recognition (OCR) is an extensive field of research in image processing, that explores the recognition and conversion of handwritten textual information to machine-encoded text, such that this information could be further processed and analyzed. Studies such as Shaikh et al. (2019) [31], utilizes OCR-based methods, combined with Convolutional Neural Networks(CNN) in auto-scoring structured handwritten answer sheets of multiple choice questions. Other studies like [34] propose an automated scoring system for handwritten student essays in reading comprehension tests, utilizing handwriting recognition and machine learning-based automated essay scoring methods. Khuong et. al [20] in their work proposes clustering handwritten mathematical answers scanned from paper-based exams, to improve the efficiency of human raters in scoring these answer sheets. Another study from Gold et. al [15], in their attempt to auto-score handwritten answers, presents the challenges of using handwriting in intelligent tutoring systems. Further, they present, how the lack of better recognition systems in these cases leads to poor scoring performances.

Recent advancements in the areas of deep learning and computer vision have led to the development of large-scale models of image representation and classification. ImageNet [10] is a large-scale image dataset widely used for training and evaluating computer vision models. Trained over 14 million images belonging to more than 22,000 different classes, ImageNet is considered a benchmark for image classification tasks. CLIP (Contrastive Language-Image Pre-training) [27] is a recently introduced image classification model based on transformer architecture, commonly used in natural language processing tasks. This method is able to encode both natural languages (text) and images in the same vector space by using a multi-modal pre-training approach. The proposed methods in this work utilizes the CLIP model to represent image and text-based answers.

2.3 The SBERT-Canberra Model

This work utilizes an auto-scoring method developed through several prior works [4, 7], referred to as the SBERT-Canberra model. As illustrated in Figure 1, the method produces a predicted score, $score_{A_s}$, for a new student answer, A_n, by leveraging the single-most-similar historic student answer, A_s. The method utilizes Sentence-BERT [28] to first generate a 768-valued feature vector for both A_n as well as all teacher-scored historic student answers, $A_{0 \ldots n-1}$ before then making a full pairwise comparison of A_n to these historic answers using Canberra distance[19]; Canberra distance is a rank-order-based distance measure that was found to more closely align to how teachers identify similarity in comparison to other distance measures such as Euclidean and Cosine Similarity [7]. From this, A_s is identified and its teacher-given score is used as the prediction for A_n; the method, therefore, adopts a variation of K-Nearest-Neighbors and has exhibited notable performance when evaluated compared to a range of baseline models [4, 13], despite its simplicity.

Through prior work, several weaknesses of the auto-scoring method have also been identified by means of a multi-level regression-based error analysis [4]. From this, four primary areas of weakness were identified: 1) model error varied greatly from problem to problem, 2) there seemed to be variation in teacher grading, 3) the presence of numbers, expressions, and equations in textual explanations correlated with higher error, and 4) the presence of images in student answers correlated with higher error. Subsequent follow-up works have explored three out of these four weaknesses, examining how answers from similar problems can be leveraged to improve predictive power for problems with smaller sample sizes [30], explore the contextual factors that contribute to variance in teacher grading practices [16], and leverage the most-frequent mathematic terms, numbers, and expressions to reduce modeling error [5]. Following these works, this paper seeks to address the fourth weakness by exploring potential methods of representing both textual and image data within similar embedding spaces.

3. DATASET

In this study, we utilize a dataset of student open-ended answers in mathematics from the prior studies [4], to compare directly with the prior works. This dataset consists of 150,477 students' answers to 2,076 different open-ended mathematics problems and scores given by 970 different teachers to these responses. The scores given by teachers to these responses are on an ordinal 5-point scale ranging from 0 to 4. The student responses given to these math-based questions are typically seen as a combination of textual responses (typed directly into the learning platform), mathematical expressions and equations, and images uploaded as a part of their work. The current dataset includes 3712 image responses in total to 311 different math problems. Some example image responses given by students are presented in Figure 2. As seen from these examples, the image-based student answers are of different types – some are handwritten, whereas others are digitally drawn images. In addition to this, these images can include handwritten text, diagrams, and graphs on a piece of paper. We can see lots of variations in these responses, in both text and image format.

4. METHODOLOGY

Utilizing the dataset from [4] and a similar model design to auto-scoring student open-response answers, we propose an extension to this prior work to support image-based responses. Similar to [4], we train a separate model per problem and perform a 10-fold cross-validation for training. For the problems without any training data, a default model based on word counts, trained across all problem data is used similarly to the prior works. In this paper, we explore and compare three different methods which we describe in detail in the following sections.

4.1 CLIP-Text Method

As stated earlier, the prior works [4], is a similarity ranking-based method, that first converts each student's answers to a 768-valued vector representation using Sentence-BERT[28], and compares answers using this vector representation and Canberra distance[19]. In our current method, we use a similar model structure with a different embedding method. This method is based on CLIP (Contrastive Language–Image Pre-training)[27] for encoding textual responses.

In the first method which we call the 'CLIP-Text' Method, we perform a text comparison similar to the prior SBERT-Canberra model, without accounting for image-based responses. Using the CLIP[27] model, we first embed the textual responses ignoring all the image responses. For any new answer in the test dataset, we compare them with the training set, by first generating a vector representation, and then comparing the vectors using Canberra distance to find the most similar pair of text responses. Using the most similar text, we utilize the score given by teachers to this similar response, in suggesting a score for the new response. In the CLIP-Text Method, we ignore the images, as we want to see how well the CLIP model does with just the text responses to directly compare it to the prior method. For any empty student responses, the model assigns a score of '0', and also for responses with no textual answers (images are discarded in this method, so if a response contains only an image, it is assigned a score of 0).

4.2 CLIP-Image Method

The second method which we call 'CLIP-Image' method, addresses both images and text in student responses. This method is similar to the 'CLIP-Text' method, with the addition of image embeddings in comparing the similarity of responses. The CLIP model uses separate text and image encoders and allows embedding text and images into the same vector space. With the CLIP model, we first encode textual and image responses into a vector representation. If a student response contains both text and images, the text part is discarded and just the images are encoded in this method. Once all the responses in the training data are encoded, for a new student answer (with either image or text-based response), its corresponding encoding is calculated and compared to the embeddings in the training

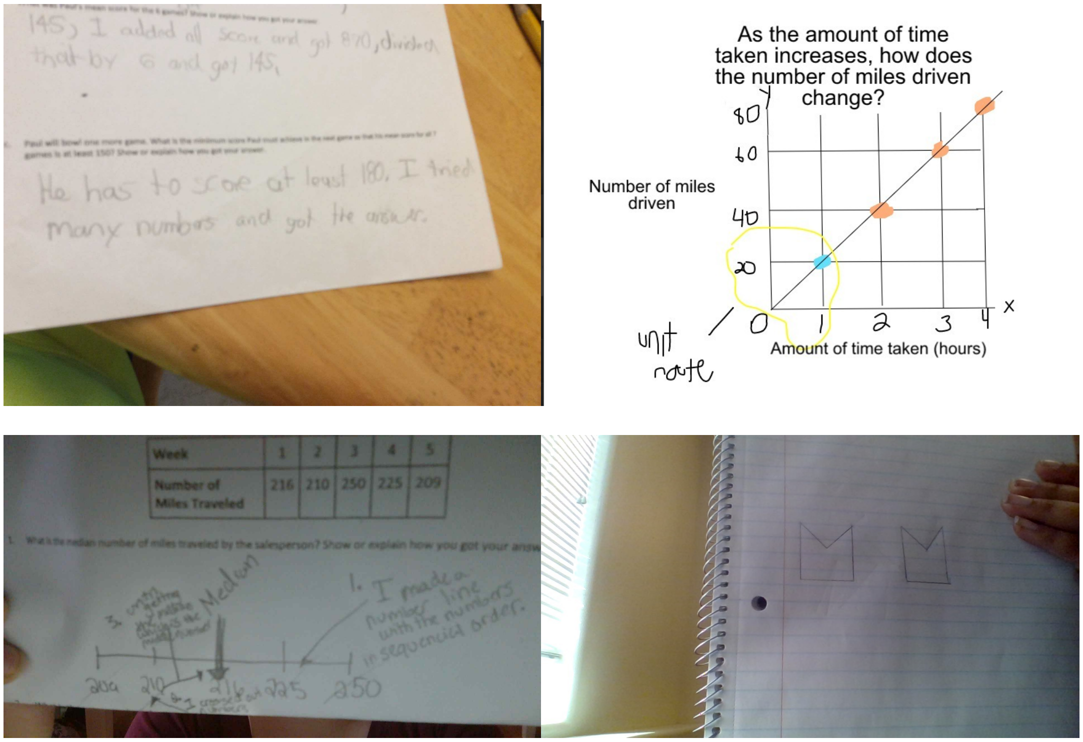

145, I added all score and got 870, divided that by 6 and got 145,
He has to score at least 180. I tried many numbers and got the answer.
As the amount of time taken increases, how does the number of miles driven change?
Number of miles driven
80
60
40
20
0
1
2
3
4
x
y
Amount of time taken (hours)
unit rate
Week
1
2
3
4
5
Number of Miles Traveled
216
210
250
225
209
Median
1. I made a number line with the numbers in sequencial order.
209
210
216
225
250

Figure 2: Examples of image-based responses from students given in response to Open-ended math problems

data, and the most similar response is selected based on the shortest Canberra distance between the new response and the responses in the training set.

4.3 CLIP-OCR Method

The third method is called 'CLIP-OCR' method which is based on state-of-the-art Optical Character Recognition (OCR). This method uses the Tesseract engine[33] from Google for text extraction. Tesseract is an open-source OCR engine, that extracts both printed and written text from images. Similar to the 'CLIP-Text', this method, then encodes the original textual responses, and also the extracted text from images (without completely ignoring the image responses). The text information from the responses is then encoded using the CLIP model, and finally, any new response is compared to the historic responses in the training data using the encodings and Canberra distance, to get a score prediction.

5. RESULTS

To answer our first and second research questions, we compare the current approaches directly to the prior methods from [13, 4]. We utilize similar evaluation methods, using a Rasch model[36] [1] that is equivalent to a traditional item response theory (IRT) model. This model aims to determine distinct parameters for each student and problem, representing student ability and problem difficulty, respectively. The rationale behind using this model is to allow a fairer comparison that accounts for factors external to the observed student response, such that the automated scoring model is evaluated solely based on its capacity to interpret the text in each student's response.

We evaluate the methods using three different metrics – AUC score, Root Mean Squared Error (RMSE), and multi-class Cohen's Kappa. The AUC score here is calculated as an average AUC over each score category and Root Mean Squared Error(RMSE) is calculated using the model estimates as a continuous-valued integer scale. The results of three methods as compared to the prior works [4] are presented in Table 1.

The result suggests that the CLIP-Text that uses the sentence embeddings from OpenAI CLIP model [27] has an AUC score of 0.852, RMSE error of 0.594, and Kappa of 0.469. Though the model doesn't outperform the prior SBERT-Canberra method [4] of auto-scoring, the difference in each of the scores is very small. The next method CLIP-Image, which compares both sentence and image embeddings using the OpenAI CLIP model, outperforms the CLIP-Text method across all three evaluation metrics used (though the difference in these scores is minimal). This method has an AUC score of 0.854, RMSE error of 0.587, and Kappa of 0.469. The next method CLIP-OCR, based on text extraction from images using OCR methods, has a similar performance to the CLIP-Image model. Though the newly introduced methods do not outperform the prior text-based method, the introduction of auto-scoring image responses is something novel that this work explores. And we can see improved performance with the addressing content from image-response in the CLIP-Image and CLIP-OCR model, than solely using text-based responses in the CLIP-Text model.

[1] A detailed description on the use of the Rasch model can be found in our prior works relating to [13, 4]

Table 1: Model Performance compared to the auto-scoring methods developed in the prior works [4]

Model	AUC	RMSE	Kappa
Current Paper			
Rasch* + CLIP-Text	0.852	0.594	0.469
Rasch* + CLIP-Image	0.854	0.587	0.471
Rasch* + CLIP-OCR	0.854	0.588	0.471
Prior works[4]			
Baseline Rasch	0.827	0.709	0.370
Rasch* + Random Forest	0.850	0.615	0.430
Rasch* + SBERT-Canberra	0.856	0.577	0.476

*These rasch models also included the number of words.

Table 2: The resulting model coefficients for the linear regression model of error for the auto-scoring method, conducted as a part of the error analysis similar to the prior method from Baral et. al [4].

	CLIP-Text		CLIP-Image		CLIP-OCR	
	B	Std. Error	B	Std. Error	B	Std. Error
Intercept	0.379***	0.005	0.361***	0.005	0.361***	0.005
Length of Answer	0.002***	0.000	0.002***	0.000	0.002***	0.000
Avg. Word Length	0.012***	0.001	0.015***	0.001	0.015***	0.001
Numbers Count	0.0002***	0.000	0.0002***	0.000	0.0002***	0.000
Operators Count	-0.001**	0.000	-0.001**	0.000	-0.001**	0.000
Equation Percent	0.139***	0.008	0.158***	0.008	0.156***	0.008
Presence of Images	2.418***	0.018	0.472***	0.018	0.560***	0.018

*p <0.05 **p<0.01 ***p<0.001;

6. ERROR ANALYSIS

To answer our third research question and to explore if the proposed image-supporting methods lead to improvements in the model's performance in the presence of images, we conduct an error analysis of the proposed methods. As previously introduced, prior work conducted an error analysis to understand the limitations of the SBERT-Canberra method [4]. This error analysis involved the calculation of several student answer-level features and using a linear regression analysis with the absolute prediction error (absolute difference between the teacher-provided score and the prediction from the model) as the dependent variable. This analysis reported that the largest amount of error in the SBERT-Canberra model was correlated with the presence of mathematical terms and equations and the presence of images in the answer text.

In this paper, we propose a method to auto-score responses in the presence of both text and images. Although the proposed methods do not outperform the previous method on auto-scoring strictly text-based answers, we hypothesize that this could be a result of using a different method of embedding text; there may be an inherent trade-off where performance is reduced for textual responses but results in improved performance where there are images (averaging out to little-to-no overall improvement). Also, from the results, we have seen improvements in the performance of the 'CLIP-Image' and 'CLIP-OCR' methods (that addresses the content of the image when auto-scoring) over the 'CLIP-Text' method (which is just based on text responses). To further study the factors that contribute to the error of these models, and to verify whether introducing image components in the text-based models actually improve the performance in the presence of images, we replicate the error analysis from Baral et. al [4]. Using features from student answers including 'Length of answer', 'Average word length', 'Total numbers count', 'Total operators', 'Percentage of equations' and 'Presence of images' as the dependent variables and Absolute model error as the independent variable, we perform three different linear regression analyses corresponding to the three proposed methods for auto-scoring.

6.1 Results of Error Analysis

The results of the error analysis are presented in Table 2. All the features from student answers are statistically significant in predicting the modeling error in all three proposed methods. However, most of these features have low coefficient values, suggesting a relatively small effect, with the exception of 'Equation Percent' and 'Presence of Images' which are positively correlated with the model error in all three cases. This is similar to the results of error analysis from prior study [4]. For the 'CLIP-Text' model, the coefficient for the presence of images is 2.418, suggesting that the presence of images in answers attributes to a notable amount of error in the model prediction, even when considering the difference in feature scaling. However, the coefficient value decreases to 0.472 in the 'CLIP-Image' method, and 0.560 in the 'CLIP-OCR' method. This decrease suggests that the introducing image component to the 'CLIP-Text' method using embedding and OCR-based text extraction actually helped the model improve in the presence of images. It is also important to note that this work does not explicitly address mathematical terms (including numbers, expressions, and equations) in the score prediction as has been suggested by other work [5]. Also, we see a slight increase in the coefficient values for equation percentage from 'CLIP-Text' to 'CLIP-Image' and 'CLIP-OCR'. For the 'CLIP-Text' method, we discard any images from the answer text, whereas for the other two methods, if there is a response that contains both image and text we discard the text from these responses and just consider the images. The change in the coefficient values for equation percent could be a result of this quality.

Following the error analysis procedure introduced in [4], we additionally applied a multi-level model to examine model error while accounting for clustered variance at the teacher-, problem-, and student-levels. We used a similar regression model with answer-level variables at level 1 (i.e. those listed in Table 2) and teacher-, problem-, and student- identifiers at level 2 as random effects. As before we again observed model error as the dependent variable. Controlling for these additional random effects did not lead to differences in the interpretation of our results; for this reason, we have omitted these additional regression results due to space limitations.

7. LIMITATIONS AND FUTURE WORKS

This paper represents an initial step toward improving state-of-the-art methods for auto-scoring student responses to mathematical problems in the presence of images. This is a preliminary work conducted towards exploring the feasibility and challenges in auto-scoring student image responses in the mathematical domain. Thus, the methods presented have several limitations and challenges that can be addressed with future work.

The proposed methods in this work use the CLIP model [27] trained on a large variety of datasets of images and natural language available over the internet. While this method shows promising results in recognizing a range of common objects, the pre-trained model may not have been exposed to the dataset of student hand-written or hand-drawn mathematics; the model was trained for application in very broad domains to recognize objects and is not optimized for identifying similar responses on paper. It has also been found that while the CLIP model learns a capable OCR system, it exhibits low accuracy in the case of handwritten digits in the widely-used MNIST dataset [27]. Further, fine-tuning this model on a mathematical dataset could lead to better model performance.

It is also important to note that the OCR method is based on the Tesseract [33] engine; this is rather a traditional OCR method and more recent advancements in OCR technology may be explored in the future to achieve improved results. Additionally, this method is known to be sensitive to poor-quality images, complex backgrounds, variation in handwriting styles, and ambiguity in the characters [33]. All of these are the common qualities of the images found in our dataset. While this method supports digital images (that are screenshots of work done on a computer), the method has low accuracy in extracting textual information from handwritten answers. Thus, exploring better OCR methods that support both handwritten and digital textual answers would better improve these auto-scoring methods for images. Further, both of the proposed methods that support images, inherently discard the additional text if present in the response. These texts may present additional supporting information to the image-based answers, so it is important to explore how to address this when evaluating these responses.

Apart from the limitation mentioned above, the process of analyzing and processing these image-based answers in itself is a challenging task, as we can see a lot of variation in these images of student-provided answers. Figure 2, presents some examples of image-based student answers. The student work in these images are not always clearly presented and structured – some handwriting is hard to read, the images sometimes are of low resolution and are blurry, the use of pencils makes the writing feint and hard to read, and lacks consistent formatting. Due to the freedom provided to students by the use of paper and pencil to draw out their solution, the resulting answer is not always structured in the same way from student to student. Future work could help address some of these challenges by implementing a more rigorous cleaning and preprocessing procedure prior to applying any image representation models. Cropping images to focus on the prominent aspects of student work, rotating images to improve the consistency of orientation, and even color correction can help improve the clarity of the work.

In all of this work, there are also several ethical concerns that should be considered in developing and applying these various methods. Images may contain Personally Identifiable Information(PII) such as students' names, faces, skin color, etc. which exposes a potential risk of biases or disparate performance in regard to the machine learning models. Future works could mitigate some of these challenges by utilizing some of the pre-processing methods described above, but also emphasizes the importance of evaluating these scoring models for potential biases or unfairness in their predictions.

8. CONCLUSION

In this study, we have presented preliminary work towards developing an auto-scoring method for student response in mathematics that includes images. By building upon the prior research in auto-scoring text-based mathematical answers, we have proposed methods for representing and scoring image-based responses. In addressing our first research question, our proposed methods did not outperform the current state-of-the-art approach for auto-scoring, but they did exhibit comparable performance across all three evaluation metrics used. Addressing our second research question, we did not find meaningful differences between the different image-supporting embedding methods. The results of the conducted error analysis, in alignment with our third research question, further indicate that using pre-existing methods of text and image embeddings can enhance the performance of the auto-scoring models in the presence of images. Our findings from this study point toward new directions for research in the area of analyzing and processing image-based student responses in mathematics.

9. ACKNOWLEDGMENTS

We thank multiple grants (e.g., 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, 1535428, 1440753, 1316736, 1252297, 1109483, & DRL-1031398); IES R305A170137, R305A170243, R305A180401, R305A120125, R305A180401, & R305C100024, P200A180088 & P200A150306, as well as N00014-18-1-2768, Schmidt Futures and a second anonymous philanthropy.

10. REFERENCES

[1] L. K. Allen, M. E. Jacovina, and D. S. McNamara. Computer-based writing instruction. *Grantee Submission*, 2016.

[2] P. An, K. Holstein, B. d'Anjou, B. Eggen, and S. Bakker. The ta framework: Designing real-time teaching augmentation for k-12 classrooms. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2020.

[3] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier. Cognitive tutors: Lessons learned. *The journal of the learning sciences*, 4(2):167–207, 1995.

[4] S. Baral, A. F. Botelho, J. A. Erickson, P. Benachamardi, and N. T. Heffernan. Improving automated scoring of student open responses in mathematics. *International Educational Data Mining Society*, 2021.

[5] S. Baral, K. Seetharaman, A. F. Botelho, A. Wang, G. Heineman, and N. T. Heffernan. Enhancing auto-scoring of student open responses in the presence of mathematical terms and expressions. In *International Conference on Artificial Intelligence in Education*, pages 685–690. Springer, 2022.

[6] S. Bhatnagar, N. Lasry, M. Desmarais, and E. Charles. Dalite: Asynchronous peer instruction for moocs. In *European Conference on Technology Enhanced Learning*, pages 505–508. Springer, 2016.

[7] A. Botelho, S. Baral, J. A. Erickson, P. Benachamardi, and N. T. Heffernan. Leveraging natural language processing to support automated assessment and feedback for student open responses in mathematics. *Journal of Computer Assisted Learning*, 2023.

[8] J. Burstein, J. Tetreault, and N. Madnani. The e-rater® automated essay scoring system. In *Handbook of automated essay evaluation*, pages 77–89. Routledge, 2013.

[9] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[11] P. Denny, A. Luxton-Reilly, and J. Hamer. Student use of the peerwise system. In *Proceedings of the 13th annual conference on Innovation and technology in computer science education*, pages 73–77, 2008.

[12] D. Ebert. Graphing projects with desmos. *The Mathematics Teacher*, 108(5):388–391, 2014.

[13] J. A. Erickson, A. F. Botelho, S. McAteer, A. Varatharaj, and N. T. Heffernan. The automated grading of student open responses in mathematics. In *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, pages 615–624, 2020.

[14] M. Fowler, B. Chen, S. Azad, M. West, and C. Zilles. Autograding" explain in plain english" questions using nlp. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, pages 1163–1169, 2021.

[15] C. Gold and T. Zesch. Exploring the impact of handwriting recognition on the automated scoring of handwritten student answers. In *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 252–257. IEEE, 2020.

[16] A. Gurung, A. F. Botelho, R. Thompson, A. C. Sales, S. Baral, and N. T. Heffernan. Considerate, unfair, or just fatigued? examining factors that impact teacher. In *Proceedings of the 30th International Conference on Computers in Education.*, 2022.

[17] N. T. Heffernan and C. L. Heffernan. The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497, 2014.

[18] M. Hohenwarter and M. Hohenwarter. Geogebra. *Available on-line at http://www. geogebra. org/cms/en*, 2002.

[19] G. Jurman, S. Riccadonna, R. Visintainer, and C. Furlanello. Canberra distance on ranked lists. In *Proceedings of advances in ranking NIPS 09 workshop*, pages 22–27. Citeseer, 2009.

[20] V. T. M. Khuong, H. Q. Ung, C. T. Nguyen, and M. Nakagawa. Clustering offline handwritten mathematical answers for computer-assisted marking. In *Proc. 1st Int. Conf. on Pattern Recognit. and Artificial Intelligence, Montreal, Canada*, pages 121–126, 2018.

[21] Y.-S. G. Kim, C. Schatschneider, J. Wanzek, B. Gatlin, and S. Al Otaiba. Writing evaluation: rater and task effects on the reliability of writing scores for children in grades 3 and 4. *Reading and writing*, 30(6):1287–1310, 2017.

[22] A. S. Lan, D. Vats, A. E. Waters, and R. G. Baraniuk. Mathematical language processing: Automatic grading and feedback for open response mathematical questions. In *Proceedings of the second (2015) ACM conference on learning@ scale*, pages 167–176, 2015.

[23] K. Leelawong and G. Biswas. Designing learning by teaching agents: The betty's brain system. *International Journal of Artificial Intelligence in Education*, 18(3):181–208, 2008.

[24] A. Mitrovic. An intelligent sql tutor on the web. *International Journal of Artificial Intelligence in Education*, 13(2-4):173–197, 2003.

[25] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. *Advances in neural information processing systems*, 28, 2015.

[26] T. Price, R. Zhi, and T. Barnes. Evaluation of a data-driven feedback algorithm for open-ended programming. *International Educational Data Mining Society*, 2017.

[27] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, volume 2, pages 8748–8763. PMLR, 2021.

[28] N. Reimers and I. Gurevych. Sentence-bert: Sentence

embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[29] B. Riordan, A. Horbach, A. Cahill, T. Zesch, and C. Lee. Investigating neural architectures for short answer scoring. In *Proceedings of the 12th workshop on innovative use of NLP for building educational applications*, pages 159–168, 2017.

[30] R. Rivera-Bergollo, S. Baral, A. Botelho, and N. Heffernan. Leveraging auxiliary data from similar problems to improve automatic open response scoring. *Proceedings of the 15th International Conference on Educational Data Mining*, pages 679–683, 2022.

[31] E. Shaikh, I. Mohiuddin, A. Manzoor, G. Latif, and N. Mohammad. Automated grading for handwritten answer sheets using convolutional neural networks. In *2019 2nd International conference on new trends in computing sciences (ICTCS)*, pages 1–6. Ieee, 2019.

[32] J. T. Shen, M. Yamashita, E. Prihar, N. Heffernan, X. Wu, B. Graff, and D. Lee. Mathbert: A pre-trained language model for general nlp tasks in mathematics education. *arXiv preprint arXiv:2106.07340*, 2021.

[33] R. Smith. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE, 2007.

[34] S. Srihari, J. Collins, R. Srihari, P. Babu, and H. Srinivasan. Automated scoring of handwritten essays based on latent semantic analysis. In *International Workshop on Document Analysis Systems*, pages 71–83. Springer, 2006.

[35] J. B. Wiggins, K. E. Boyer, A. Baikadi, A. Ezen-Can, J. F. Grafsgaard, E. Y. Ha, J. C. Lester, C. M. Mitchell, and E. N. Wiebe. Javatutor: an intelligent tutoring system that adapts to cognitive and affective states during computer programming. In *Proceedings of the 46th acm technical symposium on computer science education*, pages 599–599, 2015.

[36] B. D. Wright. Solving measurement problems with the rasch model. *Journal of educational measurement*, pages 97–116, 1977.

[37] L. Zhang, Y. Huang, X. Yang, S. Yu, and F. Zhuang. An automatic short-answer grading model for semi-open-ended questions. *Interactive learning environments*, 30(1):177–190, 2022.

[38] M. Zhang, S. Baral, N. Heffernan, and A. Lan. Automatic short math answer grading via in-context meta-learning. *arXiv preprint arXiv:2205.15219*, 2022.

[39] S. Zhao, Y. Zhang, X. Xiong, A. Botelho, and N. Heffernan. A memory-augmented neural model for automated grading. In *Proceedings of the fourth (2017) ACM conference on learning@ scale*, pages 189–192, 2017.

Generating High-Precision Feedback for Programming Syntax Errors using Large Language Models*

Tung Phung[1]
MPI-SWS
mphung@mpi-sws.org

José Cambronero[2]
Microsoft
jcambronero@microsoft.com

Sumit Gulwani[2]
Microsoft
sumitg@microsoft.com

Tobias Kohn[2]
TU Wien
tobias.kohn@tuwien.ac.at

Rupak Majumdar[2]
MPI-SWS
rupak@mpi-sws.org

Adish Singla[2]
MPI-SWS
adishs@mpi-sws.org

Gustavo Soares[2]
Microsoft
gsoares@microsoft.com

ABSTRACT

Large language models (LLMs), such as Codex, hold great promise in enhancing programming education by automatically generating feedback for students. We investigate using LLMs to generate feedback for fixing syntax errors in Python programs, a key scenario in introductory programming. More concretely, given a student's buggy program, our goal is to generate feedback comprising a fixed program along with a natural language explanation describing the errors/fixes, inspired by how a human tutor would give feedback. While using LLMs is promising, the critical challenge is to ensure high precision in the generated feedback, which is imperative before deploying such technology in classrooms. The main research question we study is: *Can we develop LLMs-based feedback generation techniques with a tunable precision parameter, giving educators quality control over the feedback that students receive?* To this end, we introduce PyFiXV, our technique to generate high-precision feedback powered by Codex. The key idea behind PyFiXV is to use a novel run-time validation mechanism to decide whether the generated feedback is suitable for sharing with the student; notably, this validation mechanism also provides a precision knob to educators. We perform an extensive evaluation using two real-world datasets of Python programs with syntax errors and show the efficacy of PyFiXV in generating high-precision feedback.

Keywords

Programming education, Python programs, syntax errors, feedback generation, large language models

*[1]: Corresponding author.
[2]: Listed in alphabetical order.

T. Phung, J. Cambronero, S. Gulwani, T. Kohn, R. Majumdar, A. Singla, and G. Soares. Generating high-precision feedback for programming syntax errors using large language models. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 370–377, Bengaluru, India, July 2023. International Educational Data Mining Society.

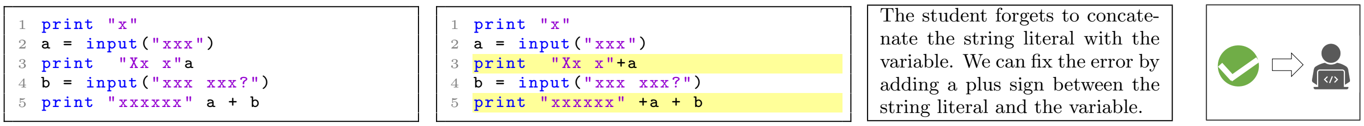
https://doi.org/10.5281/zenodo.8115653

1. INTRODUCTION

Large language models (LLMs) trained on text and code have the potential to power next-generation AI-driven educational technologies and drastically improve the landscape of computing education. One of such popular LLMs is OpenAI's Codex [1], a variant of the 175 billion parameter model GPT-3 [2], trained by fine-tuning GPT-3 on code from over 50 million GitHub repositories. A recent study ranked Codex in the top quartile w.r.t. students in a large introductory programming course [3]. Subsequently, recent works have shown promising results in using Codex on various programming education scenarios, including generating new programming assignments [4], providing code explanations [5], and enhancing programming-error-messages [6].

We investigate the use of LLMs to generate feedback for programming syntax errors, a key scenario in introductory programming education. Even though such errors typically require small fixes and are easily explainable by human tutors, they can pose a major hurdle in learning for novice students [7]. Moreover, the programming-error-messages provided by the default programming environment are often cryptic and unable to provide explicable feedback to students [8–10]. Ideally, a human tutor would help a novice student by providing detailed feedback describing the errors and required fixes to the buggy program; however, it is extremely tedious/challenging to provide feedback at scale given the growing enrollments in introductory programming courses [11, 12]. To this end, our goal is to automate the feedback generation process using LLMs-based techniques.

More concretely, given a student's buggy program, we want to generate feedback comprising a fixed program and a natural language explanation describing the errors/fixes, inspired by how a human tutor would give feedback. While models like Codex, trained on both text and code, are naturally suitable for this, the critical challenge is to ensure high precision in the generated feedback. High precision is imperative in building educators' trust before deploying such an AI-driven technology in classrooms. A recent work investigated enhancing the default programming-error-messages using Codex [6]; one of the takeaways, quoted from their paper, is *"The key implications of this work are that program-*



```
print "x"
a = input("xxx")
print  "Xx x"a
b = input("xxx xxx?")
print "xxxxxx" a + b
```

(a) Student's buggy program

```
print "x"
a = input("xxx")
print  "Xx x"+a
b = input("xxx xxx?")
print "xxxxxx" +a + b
```

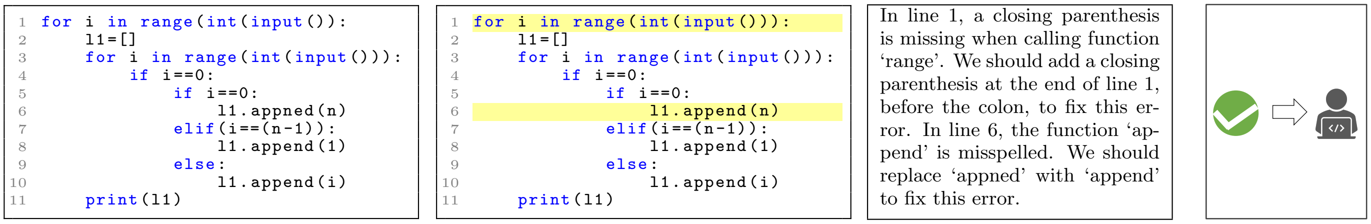
(b) Generated fixed program

The student forgets to concatenate the string literal with the variable. We can fix the error by adding a plus sign between the string literal and the variable.

(c) Generated explanation

(d) Validation

Figure 1: Illustrative example showcasing PyFiXV on a buggy Python 2 program from TigerJython [9]. **(a)** Buggy program (with string literals anonymized as sequences of 'x' in our dataset). **(b, c)** Fixed program and explanation generated by PyFiXV. **(d)** PyFiXV's validation mechanism *successfully accepted* the generated feedback as suitable for sharing.



```
1  for i in range(int(input()):
2      l1=[]
3      for i in range(int(input())):
4          if i==0:
5              if i==0:
6                  l1.appned(n)
7              elif(i==(n-1)):
8                  l1.append(1)
9              else:
10                 l1.append(i)
11     print(l1)
```

(a) Student's buggy program

```
1  for i in range(int(input())):
2      l1=[]
3      for i in range(int(input())):
4          if i==0:
5              if i==0:
6                  l1.append(n)
7              elif(i==(n-1)):
8                  l1.append(1)
9              else:
10                 l1.append(i)
11     print(l1)
```

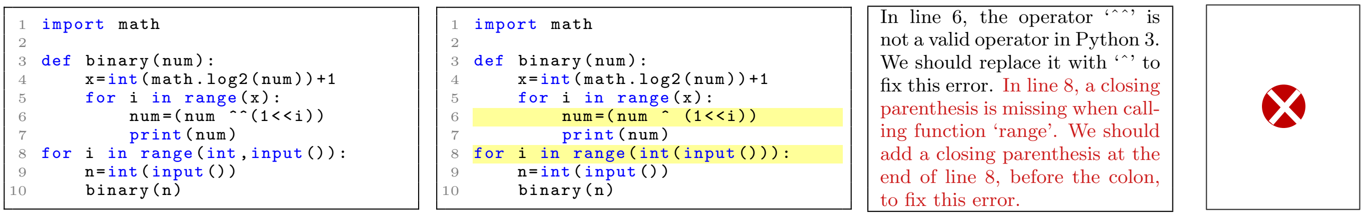
(b) Generated fixed program

In line 1, a closing parenthesis is missing when calling function 'range'. We should add a closing parenthesis at the end of line 1, before the colon, to fix this error. In line 6, the function 'append' is misspelled. We should replace 'appned' with 'append' to fix this error.

(c) Generated explanation

(d) Validation

Figure 2: Similar to Figure 1, this example showcases PyFiXV on a buggy Python 3 program from Codeforces [13]. While the misspelling of 'append' as 'appned' is not a syntax error (raising an AttributeError at runtime instead), PyFiXV typically fixes such mistakes as well, showing the generalization value in LLMs-powered technique.



```
1  import math
2
3  def binary(num):
4      x=int(math.log2(num))+1
5      for i in range(x):
6          num=(num ^^(1<<i))
7          print(num)
8  for i in range(int,input()):
9      n=int(input())
10     binary(n)
```

(a) Student's buggy program

```
1  import math
2
3  def binary(num):
4      x=int(math.log2(num))+1
5      for i in range(x):
6          num=(num ^ (1<<i))
7          print(num)
8  for i in range(int(input())):
9      n=int(input())
10     binary(n)
```

(b) Generated fixed program

In line 6, the operator '^^' is not a valid operator in Python 3. We should replace it with '^' to fix this error. In line 8, a closing parenthesis is missing when calling function 'range'. We should add a closing parenthesis at the end of line 8, before the colon, to fix this error.

(c) Generated explanation

(d) Validation

Figure 3: Similar to Figure 2, this example showcases PyFiXV on a buggy Python 3 program from Codeforces [13]. PyFiXV's validation mechanism *successfully rejected* the generated feedback (we marked text in **(c)** to highlight issues with explanation).

ming error message explanations and suggested fixes generated by LLMs are not yet ready for production use in introductory programming classes...". Our initial experiments (Section 4) also highlight issues in generating high-precision feedback. To this end, the main research question is:

Can we develop LLMs-based feedback generation techniques with a tunable precision parameter, giving educators quality control over the feedback that students receive?

1.1 Our Approach and Contributions

In this paper, we develop PyFiXV, our technique to generate high-precision feedback powered by Codex. Given a student's buggy program as input, PyFiXV decomposes the overall process into (i) feedback generation (i.e., a fixed program and a natural language explanation for errors/fixes); and (ii) feedback validation (i.e., deciding whether the generated feedback is suitable for sharing with the student). One of the key ideas in PyFiXV is to use a run-time feedback validation mechanism that decides whether the generated feedback is of good quality. This validation mechanism uses Codex as a *simulated student model* – the intuition is that a good quality explanation, when provided as Codex's prompt instruction, should increase Codex's success in converting the buggy program to the fixed program. Notably, this validation also provides a tuneable precision knob to educators to control the precision and coverage trade-off. The illustrative examples in Figures 1, 2, and 3 showcase PyFiXV on three different student's buggy programs. Our main contributions are:

(I) We formalize the problem of generating high-precision feedback for programming syntax errors using LLMs, where feedback comprises a fixed program and a natural language explanation. (Section 2)

(II) We develop a novel technique, PyFiXV, that generates feedback using Codex and has a run-time feedback validation mechanism to decide whether the generated feedback is suitable for sharing. (Section 3)

(III) We perform extensive evaluations using two real-world datasets of Python programs with syntax errors and showcase the efficacy of PyFiXV. We publicly release the implementation of PyFiXV. (Section 4)[1]

[1] Github: `https://github.com/machine-teaching-group/edm2023_PyFiXV`

1.2 Related Work

Feedback generation for programming errors. There has been extensive work on feedback generation for syntactic/semantic programming errors [14–18]; however, these works have focused on fixing/repairing buggy programs without providing explanations. The work in [11] proposed a technique to generate explanations; however, it requires pre-specified rules that map errors to explanations. Another line of work, complementary to ours, has explored crowdsourcing approaches to obtain explanations provided by other students/tutors [19, 20]. There has also been extensive work on improving the programming-error-messages by designing customized environments [9, 10]. As discussed earlier, a recent study used Codex to enhance these error messages [6]; however, our work is different as we focus on generating high-precision feedback with a tuneable precision knob.

Validation of generated content. In recent work, [21] developed a technique to validate LLMs' output in the context of program synthesis. While similar in spirit, their validation mechanism is different and operates by asking LLMs to generate predicates for testing the synthesized programs. Another possible approach is to use back-translation models to validate the generated content [22, 23]; however, such a back-translation model (that generates buggy programs from explanations) is not readily available for our setting. Another approach, complementary to ours, is to use human-in-the-loop for validating low confidence outputs [24].

2. PROBLEM SETUP

Next, we introduce definitions and formalize our objective.

2.1 Preliminaries

Student's buggy program. Consider a student working on a programming assignment who has written a buggy program with syntax errors, such as shown in Figures 1a, 2a, and 3a. Formally, these syntax errors are defined by the underlying parser of the programming language [14]; we will use the Python programming language in our evaluation. Henceforth, we denote such a buggy program as $\mathcal{P}_{\text{b}}$, which is provided as an input to feedback generation techniques.

Feedback style. Given $\mathcal{P}_{\text{b}}$, we seek to generate feedback comprising a fixed program along with a natural language explanation describing the errors and fixes. This feedback style is inspired by how a human tutor would give feedback to novice students in introductory programming education [5, 9]. We denote a generated fixed program as $\mathcal{P}_{\text{f}}$, a generated explanation as $\mathcal{X}$, and generated feedback as a tuple $(\mathcal{P}_{\text{f}}, \mathcal{X})$.

Feedback quality. We assess the quality of generated feedback $(\mathcal{P}_{\text{f}}, \mathcal{X})$ w.r.t. $\mathcal{P}_{\text{b}}$ along the following binary attributes: (i) $\mathcal{P}_{\text{f}}$ is syntactically correct and is obtained by making a small number of edits to fix $\mathcal{P}_{\text{b}}$; (ii) $\mathcal{X}$ is complete, i.e., contains information about all errors and required fixes; (iii) $\mathcal{X}$ is correct, i.e., the provided information correctly explains errors and required fixes; (iv) $\mathcal{X}$ is comprehensible, i.e., easy to understand, presented in a readable format, and doesn't contain redundant information. These attributes are inspired by evaluation rubrics used in literature [6, 25–27]. In our evaluation, feedback quality is evaluated via ratings by experts along these four attributes. We measure feedback quality as binary by assigning the value of 1 (good quality) if it satisfies *all* the four quality attributes and otherwise 0 (bad quality).[2]

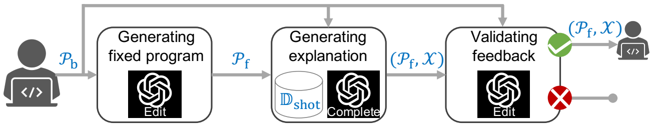


Figure 4: Illustration of three different compoments/stages in PyFiXV's feedback generation process; see Section 3.

2.2 Performance Metrics and Objective

Performance metrics. Next, we describe the overall performance metrics used to evaluate a feedback generation technique. For a buggy program $\mathcal{P}_{\text{b}}$ as input, we seek to design techniques that generate feedback $(\mathcal{P}_{\text{f}}, \mathcal{X})$ and also decide whether the generated feedback is suitable for sharing with the student. We measure the performance of a technique using two metrics: (i) *Coverage* measuring the percentage number of times the feedback is *generated and provided to the student*; (ii) *Precision* measuring the percentage number of times the *provided feedback is of good quality* w.r.t. the binary feedback quality criterion introduced above. In our experiments, we will compute these metrics on a dataset $\mathbb{D}_{\text{test}} = \{\mathcal{P}_{\text{b}}\}$ comprising a set of students' buggy programs.[3]

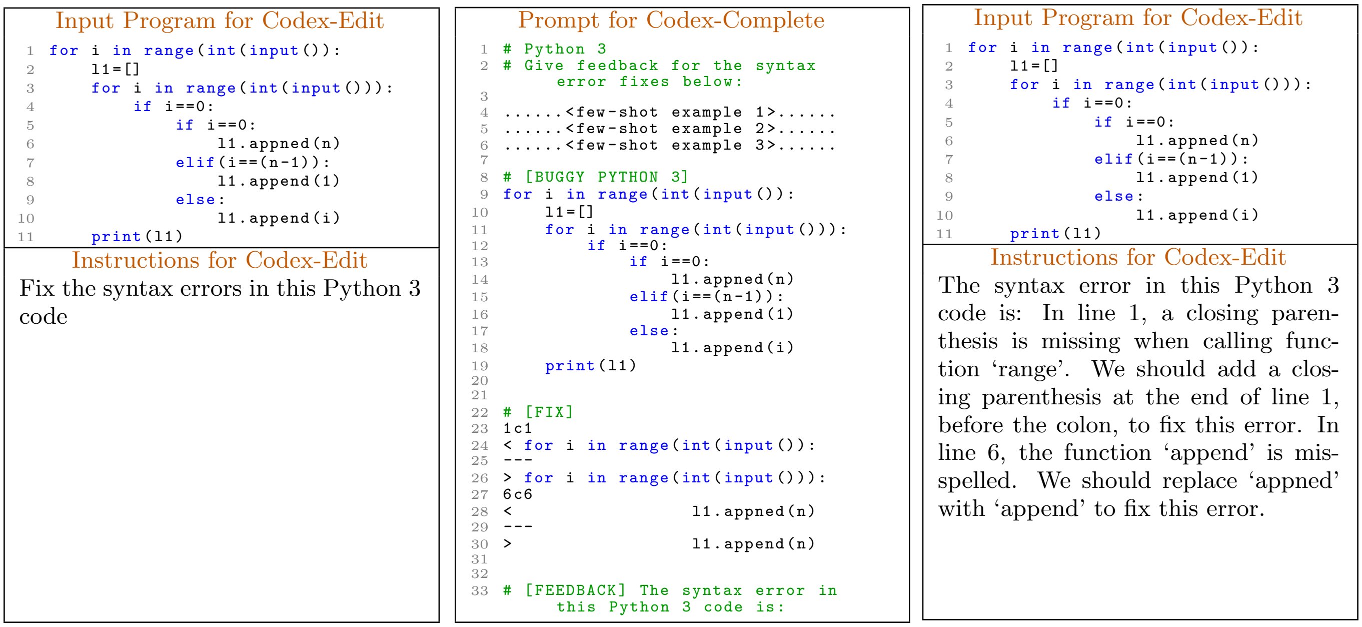
Objective. Our goal is to design feedback generation techniques with high precision, which is imperative before deploying such techniques in classrooms. In particular, we want to develop techniques with a tuneable precision parameter that could provide a knob to educators to control the precision and coverage trade-off.

3. OUR TECHNIQUE PyFiXV

In this section, we present PyFiXV, our technique to generate high-precision feedback using LLMs. PyFiXV uses OpenAPI's Codex as LLMs [1] – Codex has shown competitive performance on a variety of programming benchmarks [1, 3, 17, 18], and is particularly suitable for PyFiXV as we seek to generate both fixed programs and natural language explanations. More specifically, PyFiXV uses two access points of Codex provided by OpenAI through public APIs: Codex-Edit [28] and Codex-Complete [29]. As illustrated in Figure 4, PyFiXV has the following three components/stages: (1) generating a fixed program $\mathcal{P}_{\text{f}}$ by editing $\mathcal{P}_{\text{b}}$ using Codex-Edit; (2) generating natural language explanation $\mathcal{X}$ using Codex-Complete; (3) validating feedback $(\mathcal{P}_{\text{f}}, \mathcal{X})$ using Codex-Edit to decide whether the generated feedback is suitable for sharing. The overall pipeline of PyFiXV is modular and we will evaluate the utility of different components in Section 4. Next, we provide details for each of these stages.

[2] We note that the four attributes are independent. In particular, the attribute "complete" captures whether the explanation contains information about all errors/fixes (even though the information could be wrong), and the attribute "correct" captures the correctness of the provided information.

[3] When a technique cannot generate feedback for an input program $\mathcal{P}_{\text{b}}$ (e.g., the technique is unable to find a fixed program), then we use a natural convention that no feedback is provided to the student—this convention lowers the coverage metric but doesn't directly affect the precision metric.



Input Program for Codex-Edit

```
1  for i in range(int(input()):
2      l1=[]
3      for i in range(int(input())):
4          if i==0:
5              if i==0:
6                  l1.appned(n)
7              elif(i==(n-1)):
8                  l1.append(1)
9              else:
10                 l1.append(i)
11     print(l1)
```

Instructions for Codex-Edit

Fix the syntax errors in this Python 3 code

(a) Stage-1 prompt for generating $\mathcal{P}_{\mathrm{f}}$

Prompt for Codex-Complete

```
1  # Python 3
2  # Give feedback for the syntax
       error fixes below:
3
4  ......<few-shot example 1>......
5  ......<few-shot example 2>......
6  ......<few-shot example 3>......
7
8  # [BUGGY PYTHON 3]
9  for i in range(int(input()):
10     l1=[]
11     for i in range(int(input())):
12         if i==0:
13             if i==0:
14                 l1.appned(n)
15             elif(i==(n-1)):
16                 l1.append(1)
17             else:
18                 l1.append(i)
19     print(l1)
20
21
22 # [FIX]
23 1c1
24 < for i in range(int(input()):
25 ---
26 > for i in range(int(input())):
27 6c6
28 <                   l1.appned(n)
29 ---
30 >                   l1.append(n)
31
32
33 # [FEEDBACK] The syntax error in
       this Python 3 code is:
```

(b) Stage-2 prompt for generating $\mathcal{X}$

Input Program for Codex-Edit

```
1  for i in range(int(input()):
2      l1=[]
3      for i in range(int(input())):
4          if i==0:
5              if i==0:
6                  l1.appned(n)
7              elif(i==(n-1)):
8                  l1.append(1)
9              else:
10                 l1.append(i)
11     print(l1)
```

Instructions for Codex-Edit

The syntax error in this Python 3 code is: In line 1, a closing parenthesis is missing when calling function 'range'. We should add a closing parenthesis at the end of line 1, before the colon, to fix this error. In line 6, the function 'append' is misspelled. We should replace 'appned' with 'append' to fix this error.

(c) Stage-3 prompt for validating $(\mathcal{P}_{\mathrm{f}}, \mathcal{X})$

Figure 5: Illustration of prompts used by different stages of PyFiXV for buggy Python 3 program in Figure 2. In particular, the "Instructions for Codex-Edit" in **(c)** is obtained by concatenating line33 of **(b)** and the generated $\mathcal{X}$ shown in Figure 2c.

3.1 Stage-1: Generating Fixed Program

Given a student's buggy program $\mathcal{P}_{\mathrm{b}}$ as input, PyFiXV's Stage-1 generates a fixed program $\mathcal{P}_{\mathrm{f}}$. We use Codex-Edit for fixing/repairing the buggy program in this stage since it has shown to be competitive in program repair benchmarks in recent works [30]. Figure 5a shows a sample prompt used by PyFiXV to query Codex-Edit for the buggy Python 3 program in Figure 2a. The process of generating $\mathcal{P}_{\mathrm{f}}$ is determined by two hyperparameters: (i) $t_1 \in [0.0, 1.0]$ is the temperature value specified when querying Codex-Edit and controls stochasticity/diversity in generated programs; (ii) n_1 controls the number of queries made to Codex-Edit.

More concretely, PyFiXV begins by making n_1 queries to Codex-Edit with temperature t_1. Then, out of n_1 generated programs, PyFiXV selects $\mathcal{P}_{\mathrm{f}}$ as the program that is syntactically correct and has the smallest *edit-distance* to $\mathcal{P}_{\mathrm{b}}$. Here, edit-distance between two programs is measured by first tokenizing programs using Pygments library [31] and then computing Levenshtein edit-distance over token strings.[4] If Stage-1 is unable to generate a fixed program, the process stops without generating any feedback; see Footnote 3. In our experiments, we set $(t_1 = 0.5, n_1 = 10)$ and obtained a high success rate of generating a fixed program $\mathcal{P}_{\mathrm{f}}$ with a small number of edits w.r.t. $\mathcal{P}_{\mathrm{b}}$.

3.2 Stage-2: Generating Explanation

Given $\mathcal{P}_{\mathrm{b}}$ and $\mathcal{P}_{\mathrm{f}}$ as inputs, PyFiXV's Stage-2 generates a natural language explanation $\mathcal{X}$ describing errors/fixes. We use Codex-Complete in this stage as it is naturally suited to generate text by completing a prompt [1, 5, 6]. A crucial ingredient of Stage-2 is the annotated dataset $\mathbb{D}_{\mathrm{shot}}$ used to select few-shot examples when querying Codex-Complete (see Figure 4). Figure 5b shows a sample prompt used by PyFiXV to query Codex-Complete for the scenario in Figure 2. In Figure 5b, line4–line6 indicate three few-shot examples (not shown for conciseness), line9–line19 provides $\mathcal{P}_{\mathrm{b}}$, line23–line30 provides $\mathcal{P}_{\mathrm{f}}$ in the form of line-diff w.r.t. $\mathcal{P}_{\mathrm{b}}$, and line33 is the instruction to be completed by Codex-Complete. Given a prompt, the process of generating $\mathcal{X}$ is determined by two hyperparameters: (i) a temperature value t_2 $(= 0)$ and (ii) the number of queries n_2 $(= 1)$. Next, we discuss the role of $\mathbb{D}_{\mathrm{shot}}$ in selecting few-shots examples.

When querying Codex-Complete, we use three few-shot examples selected from $\mathbb{D}_{\mathrm{shot}}$, an annotated dataset of examples comprising buggy programs and desired feedback obtained by expert annotations (see Section 4.2). These annotated examples essentially provide a context to LLMs and have shown to play an important role in optimizing the generated output (e.g., see [1, 2, 17, 18, 32]). In our case, $\mathbb{D}_{\mathrm{shot}}$ provides contextualized training data, capturing the format of how experts/tutors give explanations. Given $\mathcal{P}_{\mathrm{b}}$ and $\mathcal{P}_{\mathrm{f}}$, we use two main criteria to select few-shot examples. The primary criterion is to pick examples where the error type of buggy program in the example is same as that of $\mathcal{P}_{\mathrm{b}}$—the underlying parser/compiler provides error types (e.g., 'InvalidSyntax', 'UnexpectedIndent'). The secondary criterion (used to break ties in the selection process) is based on the edit-distance *between* the *diff* of buggy/fixed program in the example and *diff* of $\mathcal{P}_{\mathrm{b}}/\mathcal{P}_{\mathrm{f}}$. In Section 4, we conduct ablations to showcase the importance of selecting few-shots.

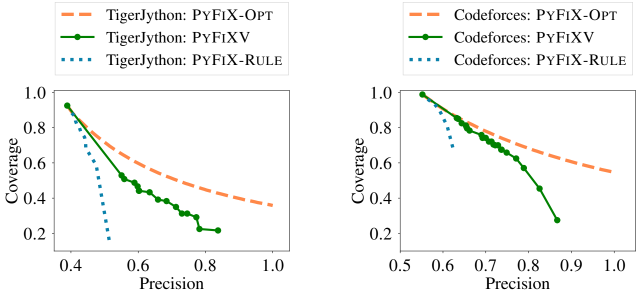
3.3 Stage-3: Validating Feedback

Given $\mathcal{P}_{\mathrm{b}}$ and $(\mathcal{P}_{\mathrm{f}}, \mathcal{X})$ as inputs, PyFiXV's Stage-3 validates the feedback quality and makes a binary decision of

[4] Note that buggy programs are not parseable to *Abstract Syntax Tree* (AST) representations and string-based distance is commonly used in such settings (e.g., see [17]).

Technique	**TigerJython**		**Codeforces**	
	Precision	Coverage	Precision	Coverage
PyFi-PEM	05.0 (1.0)	92.5 (1.6)	35.0 (2.4)	98.8 (0.8)
PyFiX$_{\text{shot:None}}$	00.9 (0.5)	92.5 (1.6)	03.0 (0.4)	98.8 (0.8)
PyFiX$_{\text{shot:Rand}}$	21.6 (1.7)	92.5 (1.6)	48.5 (2.6)	98.8 (0.8)
PyFiX$_{\text{shot:Sel}}$	38.9 (3.5)	92.5 (1.6)	55.2 (3.9)	98.8 (0.8)
PyFi$\|\|$X$_{\text{shot:Sel}}$	15.8 (1.8)	92.5 (1.6)	15.6 (2.8)	98.8 (0.8)
PyFiX-Rule$_{\text{P}\geq 70}$	48.6 (4.4)	30.8 (12.5)	61.6 (9.0)	38.3 (10.5)
PyFiXV$_{\text{P}\geq 70}$	76.0 (4.0)	31.2 (4.0)	72.4 (6.2)	64.2 (6.3)
PyFiX-Opt$_{\text{P}\approx V_{\text{P}\geq 70}}$	76.1 (0.4)	47.1 (3.4)	72.8 (0.1)	75.0 (5.7)

(a) Results for different techniques, reported as mean (stderr)



(b) TigerJython trade-off curve

(c) Codeforces trade-off curve

Figure 6: Experimental results on two real-world datasets of Python programs, namely TigerJython [9] and Codeforces [13].

"accept" (feedback is suitable for sharing) or "reject" (feedback is discarded). PyFiXV uses a novel run-time feedback validation mechanism using Codex-Edit to decide whether the feedback $(\mathcal{P}_\text{f}, \mathcal{X})$ w.r.t. $\mathcal{P}_\text{b}$ is of good quality. Here, Codex-Edit is used in the flipped role of a *simulated student model* – the intuition is that a good quality explanation $\mathcal{X}$, when provided in Codex-Edit's prompt instruction, should increase Codex-Edit's success in converting $\mathcal{P}_\text{b}$ to $\mathcal{P}_\text{f}$. Figure 5c shows a sample prompt used by PyFiXV to query Codex-Edit for the scenario in Figure 2—see the caption on how "Instructions for Codex-Edit" in Figure 5c is obtained.[5]

The validation mechanism has three hyperparameters: (i) $t_3 \in [0.0, 1.0]$ is the temperature value specified when querying Codex-Edit; (ii) n_3 controls the number of queries made to Codex-Edit; (iii) $h_3 \in [1, n_3]$ is the threshold used for acceptance decision. More concretely, PyFiXV begins by making n_3 queries to Codex-Edit with temperature t_3. Then, out of n_3 generated programs, PyFiXV counts the number of programs that don't have syntax errors and have an *exact-match* with $\mathcal{P}_\text{f}$. Here, exact-match is checked by converting programs to their *Abstract Syntax Tree* (AST)-based normalized representations.[6] Finally, the validation mechanism accepts the feedback if the number of exact matches is at least h_3. These hyperparameters (t_3, n_3, h_3) also provide a precision knob and are selected to obtain the desired precision level, as discussed next.

3.4 Precision and Coverage Trade-Off

PyFiXV's validation mechanism provides a precision knob to control the precision and coverage trade-off (see performance metrics in Section 2.2). Let P be the desired precision level we want to achieve for PyFiXV. The idea is to choose Stage-3 hyperparameters (t_3, n_3, h_3) that achieve P precision level. For this purpose, we use a calibration dataset $\mathbb{D}_\text{cal}$ for picking the hyperparameters. More concretely, in our experiments, PyFiXV first computes performance metrics on $\mathbb{D}_\text{cal}$ for the following range of values: (i) $t_3 \in \{0.3, 0.5, 0.8\}$; (ii) $n_3 \in \{10\}$; (iii) $h_3 \in \{1, 2, \ldots, 10\}$. Then, it chooses (t_3, n_3, h_3) that has at least P precision level and maximizes coverage; when achieving the desired P is not possible, then the next lower possible precision is considered. The chosen values of hyperparameters are then used in PyFiXV's Stage-3 validation mechanism. We refer to PyFiXV$_{\text{P}\geq x}$ as the version of PyFiXV calibrated with $\text{P} \geq x$.

4. EXPERIMENTAL EVALUATION

We perform evaluations using two real-world Python programming datasets, namely TigerJython [9] and Codeforces [13]. We picked Python because of its growing popularity as an introductory programming language; notably, PyFiXV can be used with other languages by appropriately changing the prompts and tokenizers used. We use OpenAI's public APIs for Codex-Edit [28] (*model=code-davinci-edit-001*) and Codex-Complete [29] (*model=code-davinci-002*). We begin by describing different techniques used in the evaluation.

4.1 Baselines and Variants of PyFiXV

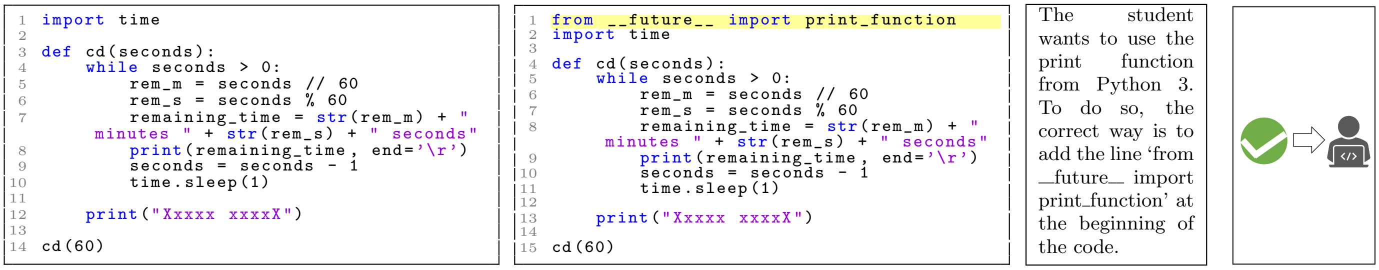
Default programming-error-messages without validation. As our first baseline, PyFi-PEM uses PyFiXV's Stage-1 to generate $\mathcal{P}_\text{f}$ and uses programming-error-messages provided by the programming environment as $\mathcal{X}$. PyFi-PEM uses error messages provided by Python 2.7 environment for TigerJython and Python 3.12 environment for Codeforces. This baseline is without validation (i.e., the generated feedback is always accepted).

Variants of PyFiXV without validation. PyFiX$_{\text{shot:Sel}}$ is a variant of PyFiXV without the validation mechanism (i.e., only uses Stage-1 and Stage-2). PyFiX$_{\text{shot:Rand}}$ is a variant of PyFiX$_{\text{shot:Sel}}$ where few-shot examples in Stage-2 are picked randomly from $\mathbb{D}_\text{shot}$. PyFiX$_{\text{shot:None}}$ is a variant of PyFiX$_{\text{shot:Sel}}$ that doesn't use few-shot examples in Stage-2. PyFi$||$X$_{\text{shot:Sel}}$ is a variant of PyFiX$_{\text{shot:Sel}}$ that runs Stage-1 and Stage-2 in parallel; hence, Stage-2's prompt doesn't make use of $\mathcal{P}_\text{f}$. All these variants are without validation (i.e., the generated feedback is always accepted).

Techniques with alternative validation mechanisms. We consider two variants of PyFiXV, namely PyFiX-Rule and PyFiX-Opt, that use different validation mechanisms (i.e., replace PyFiXV's Stage-3 with an alternative validation).

[5] In our initial experiments, we tried using alternative signals for validation, such as (a) Codex-Complete's probabilities associated with generated $\mathcal{X}$; (b) automatic scoring of $\mathcal{X}$ w.r.t. explanations in few-shots using BLEU score [33]; (c) filtering based on $\mathcal{X}$'s length. Section 4 reports results for (c) as it had the highest performance among these alternatives.

[6] We check for AST-based exact match instead of checking for Levenshtein edit-distance over token strings being 0 (see Section 3.1). AST-based exact match is more relaxed than edit-distance being 0 – AST-based representation ignores certain differences between codes, e.g., based on extra spaces and comments. We used the AST-based exact match in the validation mechanism as it is more robust to such differences.



```
import time

def cd(seconds):
    while seconds > 0:
        rem_m = seconds // 60
        rem_s = seconds % 60
        remaining_time = str(rem_m) + "
 minutes " + str(rem_s) + " seconds"
        print(remaining_time, end='\r')
        seconds = seconds - 1
        time.sleep(1)

    print("Xxxxx xxxxX")

cd(60)
```

(a) Student's buggy program $\mathcal{P}_\mathrm{b}$

```
from __future__ import print_function
import time

def cd(seconds):
    while seconds > 0:
        rem_m = seconds // 60
        rem_s = seconds % 60
        remaining_time = str(rem_m) + "
 minutes " + str(rem_s) + " seconds"
        print(remaining_time, end='\r')
        seconds = seconds - 1
        time.sleep(1)

    print("Xxxxx xxxxX")

cd(60)
```

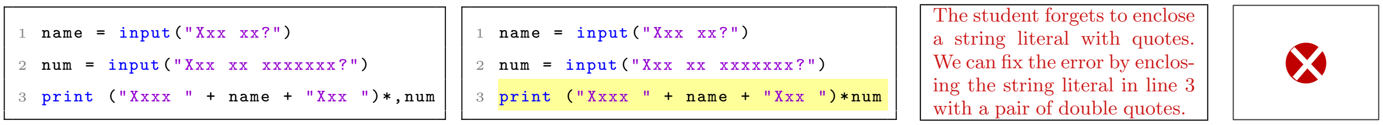
(b) Generated $\mathcal{P}_\mathrm{f}$

The student wants to use the print function from Python 3. To do so, the correct way is to add the line 'from __future__ import print_function' at the beginning of the code.

(c) Generated $\mathcal{X}$

(d) Validation

Figure 7: Similar to Figure 1, this illustrative example showcases PyFiXV on a buggy Python 2 program from TigerJython [9].



```
1 name = input("Xxx xx?")

2 num = input("Xxx xx xxxxxxx?")

3 print ("Xxxxx " + name + "Xxx ")*,num
```

(a) Student's buggy program $\mathcal{P}_\mathrm{b}$

```
1 name = input("Xxx xx?")

2 num = input("Xxx xx xxxxxxx?")

3 print ("Xxxxx " + name + "Xxx ")*num
```

(b) Generated $\mathcal{P}_\mathrm{f}$

The student forgets to enclose a string literal with quotes. We can fix the error by enclosing the string literal in line 3 with a pair of double quotes.

(c) Generated $\mathcal{X}$

(d) Validation

Figure 8: Similar to Figure 3, this example showcases PyFiXV on a buggy Python 2 program from TigerJython [9]. PyFiXV's validation mechanism *successfully rejected* the generated feedback (we marked text in **(c)** to highlight issues with explanation).

PyFiX-Rule validates $(\mathcal{P}_\mathrm{f}, \mathcal{X})$ based on $\mathcal{X}$'s length, as noted in Footnote 5. Given a hyperparameter h_r, $(\mathcal{P}_\mathrm{f}, \mathcal{X})$ is accepted if the number of tokens in $\mathcal{X}$ is at most h_r, where tokenization is done by splitting on whitespaces/punctuations. PyFiX-Rule's h_r is picked from the set $\{30, 40, 50, \ldots, 200\}$ based on the desired precision level P, by following the calibration process in Section 3.4. PyFiX-Opt uses an oracle validation that has access to expert's ratings for the generated feedback $(\mathcal{P}_\mathrm{f}, \mathcal{X})$. Then, for a desired P, PyFiX-Opt performs optimal validation and highlights the maximum coverage achievable on $\mathbb{D}_\mathrm{test}$ for the generated feedback.

4.2 Datasets and Evaluation Procedure

Datasets and annotations for few-shot examples. As our first dataset, namely TigerJython, we have 240 distinct Python 2 programs written by students in TigerJython's educational programming environment [9]. We obtained a private and anonymized version of the dataset used in [34], with string literals in programs replaced with sequences of 'x' (e.g., see Figure 1). As our second dataset, namely Codeforces, we curated 240 distinct Python 3 programs from the Codeforces website using their public APIs [13], inspired by similar works that curate Codeforces dataset [35, 36]. Programs in both datasets have syntax errors and have token length at most 500 (see Section 3.1 about program tokenization). For the Codeforces dataset, we only include programs submitted to contests held from July 2021 onwards (after the cut-off date for Codex's training data [1]). Since a part of these datasets will be used for few-shot examples (as $\mathbb{D}_\mathrm{shot}$ in PyFiXV's Stage-2), we asked experts to annotate these 480 programs with feedback (i.e., a fixed program along with an explanation). Three experts, with extensive experience in Python programming and tutoring, provided annotations.

Evaluation procedure and feedback ratings. Given a dataset $\mathbb{D}$ with 240 buggy programs, we can evaluate a technique by splitting $\mathbb{D}$ as follows: (a) $\mathbb{D}_\mathrm{test}$ (25%) for reporting precision and coverage performance metrics; (b) $\mathbb{D}_\mathrm{shot}$ (50%) for few-shot examples; (c) $\mathbb{D}_\mathrm{cal}$ (25%) for calibrating validation mechanism. To report overall performance for techniques, we perform a cross-validation procedure with four evaluation rounds while ensuring that $\mathbb{D}_\mathrm{test}$ across four rounds are non-overlapping. We then report aggregated results across these rounds as average mean (stderr). As discussed in Sections 2.1 and 2.2, evaluating these performance metrics requires feedback ratings by experts to assess the quality of the feedback generated by each technique.[7] For example, evaluating metrics on TigerJython dataset for PyFiXV requires 480 feedback ratings (4×60 for $\mathbb{D}_\mathrm{test}$ and 4×60 for $\mathbb{D}_\mathrm{cal}$). To begin, we did a smaller scale investigation to establish the rating criteria, where two experts rated 100 generated feedback instances; we obtained Cohen's kappa reliability value 0.72 indicating *substantial agreement* between experts [37]. Afterward, one expert (with experience in tutoring Python programming classes) did these feedback ratings for the evaluation results.[8]

4.3 Results

Comparison of different techniques. Figure 6a provides a comparison of different techniques on two datasets. All techniques here use PyFiXV's Stage-1 to obtain $\mathcal{P}_\mathrm{f}$. The coverage numbers of 92.5 and 98.8 reported in Figure 6a correspond to the success rate of obtaining $\mathcal{P}_\mathrm{f}$ on these datasets (the average edit-distance between $\mathcal{P}_\mathrm{b}$ and $\mathcal{P}_\mathrm{f}$ is about 10.4 and 7.5 tokens on these datasets, respectively). For our baseline PyFi-PEM, we see a big jump in precision from 5.0 for TigerJython (Python 2) to 35.0 for Codeforces (Python

[7] We note that precision and coverage performance metrics for different techniques are reported for the end-to-end process associated with each technique, and not just for the validation mechanism. Also, even if a technique doesn't use any validation mechanism, the coverage could be less than 100.0 as discussed in Footnote 3.

[8] We note that the experts were blinded to the condition (technique) associated with each feedback instance when providing ratings. Moreover, these generated feedback instances were given to experts in randomized order across conditions instead of grouping them per condition.

3), owing to enhanced error messages in recent Python versions [38–40]. Results for PyFiXV$_{\text{P}\geq 70}$ in comparison with results for PyFiX$_{\text{shot:Sel}}$, PyFiX$_{\text{shot:Rand}}$, PyFiX$_{\text{shot:None}}$, and PyFi||X$_{\text{shot:Sel}}$ showcase the utility of different components used in PyFiXV's pipeline. Comparing PyFiXV$_{\text{P}\geq 70}$ with PyFiX-Rule$_{\text{P}\geq 70}$ shows that PyFiXV's validation substantially outperforms PyFiX-Rule's validation.[9] Lastly, results for PyFiX-Opt$_{\text{P}\approx V_{\text{P}\geq 70}}$ are obtained by setting the desired precision level for PyFiX-Opt to match that of PyFiXV$_{\text{P}\geq 70}$ on $\mathbb{D}_{\text{test}}$ – the coverage numbers (47.1 for TigerJython and 75.0 for Codeforces) indicate the maximum possible achievable coverage. Notably, PyFiXV$_{\text{P}\geq 70}$ achieves a competitive coverage of 64.2 on Codeforces.[10]

Precision and coverage trade-off curves. The curves in Figures 6b and 6c are obtained by picking different desired precision levels P and then computing precision/coverage values on $\mathbb{D}_{\text{test}}$ w.r.t. P. The curves for PyFiX-Opt show the maximum possible coverage achievable on $\mathbb{D}_{\text{test}}$ for different precision levels P using our generated feedback. To obtain these curves for PyFiXV and PyFiX-Rule, we did calibration directly on $\mathbb{D}_{\text{test}}$ instead of $\mathbb{D}_{\text{cal}}$ (i.e., doing ideal calibration for their validation mechanisms when comparing with PyFiX-Opt's curves). These curves highlight the precision and coverage trade-off offered by PyFiXV in comparison to a simple rule-based validation and the oracle validation.

Qualitative analysis. We have provided several illustrative examples to demonstrate our technique PyFiXV. Figures 1, 2, and 7 show examples where PyFiXV's Stage-1 and Stage-2 generate good quality feedback and Stage-3 successfully accepts the feedback. Figures 3 and 8 show examples where PyFiXV's Stage-1 and Stage-2 generate bad quality feedback and Stage-3 successfully rejects the feedback. Figure 7 highlights that PyFiXV can make non-trivial fixes in the buggy program and correctly explain them in a comprehensible way. Figure 3 shows an example where the overall feedback is bad quality and successfully rejected, though parts of the generated explanation are correct; this could potentially be useful for tutors in a human-in-the-loop approach.

5. CONCLUDING DISCUSSIONS

We investigated using LLMs to generate feedback for fixing programming syntax errors. In particular, we considered feedback in the form of a fixed program along with a natural language explanation. We focussed on the challenge of generating high-precision feedback, which is crucial before deploying such technology in classrooms. Our proposed technique, PyFiXV, ensures high precision through a novel run-time validation mechanism and also provides a precision knob to educators. We performed an extensive evaluation to showcase the efficacy of PyFiXV on two real-world Python programming datasets. There are several interesting directions for future work, including (a) improving PyFiXV's components to obtain better precision/coverage trade-off, e.g., by adapting our technique to use recent LLMs such as ChatGPT [42] and GPT-4 [43] instead of Codex; (b) extending PyFiXV beyond syntax errors to provide feedback for programs with semantic errors or partial programs; (c) incorporating additional signals in PyFiXV's validation mechanism; (d) conducting real-world studies in classrooms.

6. ACKNOWLEDGMENTS

Funded/Co-funded by the European Union (ERC, TOPS, 101039090). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] Mark Chen and et al. Evaluating Large Language Models Trained on Code. *CoRR*, abs/2107.03374, 2021.

[2] Tom B. Brown and et al. Language Models are Few-Shot Learners. In *NeurIPS*, 2020.

[3] James Finnie-Ansley, Paul Denny, Brett A. Becker, Andrew Luxton-Reilly, and James Prather. The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming. In *ACE*, 2022.

[4] Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models. In *ICER*, 2022.

[5] Stephen MacNeil, Andrew Tran, Arto Hellas, Joanne Kim, Sami Sarsa, Paul Denny, Seth Bernstein, and Juho Leinonen. Experiences from Using Code Explanations Generated by Large Language Models in a Web Software Development E-Book. In *SIGCSE*, 2023.

[6] Juho Leinonen, Arto Hellas, Sami Sarsa, Brent N. Reeves, Paul Denny, James Prather, and Brett A. Becker. Using Large Language Models to Enhance Programming Error Messages. In *SIGCSE*, 2023.

[7] James Prather, Raymond Pettit, Kayla Holcomb McMurry, Alani L. Peters, John Homer, Nevan Simone, and Maxine S. Cohen. On Novices' Interaction with Compiler Error Messages: A Human Factors Approach. In *ICER*, 2017.

[8] Brett A. Becker. An Effective Approach to Enhancing Compiler Error Messages. In *SIGCSE*, 2016.

[9] Tobias Kohn and Bill Z. Manaris. Tell Me What's Wrong: A Python IDE with Error Messages. In *SIGCSE*, 2020.

[10] Brett A. Becker. What Does Saying That 'Programming is Hard' Really Say, and About Whom? *Communications of ACM*, 64(8):27–29, 2021.

[11] Rishabh Singh, Sumit Gulwani, and Armando Solar-Lezama. Automated Feedback Generation for Introductory Programming Assignments. In *PLDI*, 2013.

[9] When comparing PyFiXV$_{\text{P}\geq 70}$ with these techniques in Figure 6a, the results are significantly different w.r.t. χ^2 tests [41] ($p \leq 0.0001$); here, we use contingency tables with two rows (techniques) and four columns (240 data points mapped to four possible precision/coverage outcomes).

[10] Techniques PyFiX$_{\text{shot:Sel}}$, PyFiX-Rule, PyFiXV$_{\text{P}\geq 70}$, and PyFiX-Opt$_{\text{P}\approx V_{\text{P}\geq 70}}$ differ only in terms of validation mechanisms. We can compare the validation mechanisms used in these techniques based on F1-score. The F1-scores of these four techniques are as follows: 0.56, 0.39, 0.70, and 0.86 for TigerJython, respectively; 0.71, 0.47, 0.77, and 0.84 for Codeforces, respectively.

[12] Samim Mirhosseini, Austin Z. Henley, and Chris Parnin. What is Your Biggest Pain Point? An Investigation of CS Instructor Obstacles, Workarounds, and Desires. In *SIGCSE*, 2023.

[13] Mikhail Mirzayanov. Codeforces. `https://codeforces.com/`.

[14] Sumit Gulwani, Ivan Radicek, and Florian Zuleger. Automated Clustering and Program Repair for Introductory Programming Assignments. In *PLDI*, 2018.

[15] Sahil Bhatia, Pushmeet Kohli, and Rishabh Singh. Neuro-Symbolic Program Corrector for Introductory Programming Assignments. In *ICSE*, 2018.

[16] Rahul Gupta, Aditya Kanade, and Shirish K. Shevade. Deep Reinforcement Learning for Syntactic Error Repair in Student Programs. In *AAAI*, 2019.

[17] Jialu Zhang, José Cambronero, Sumit Gulwani, Vu Le, Ruzica Piskac, Gustavo Soares, and Gust Verbruggen. Repairing Bugs in Python Assignments Using Large Language Models. *CoRR*, abs/2209.14876, 2022.

[18] Harshit Joshi, José Pablo Cambronero Sánchez, Sumit Gulwani, Vu Le, Ivan Radicek, and Gust Verbruggen. Repair is Nearly Generation: Multilingual Program Repair with LLMs. In *AAAI*, 2023.

[19] Björn Hartmann, Daniel MacDougall, Joel Brandt, and Scott R. Klemmer. What Would Other Programmers Do: Suggesting Solutions to Error Messages. In *CHI*, 2010.

[20] Andrew Head, Elena L. Glassman, Gustavo Soares, Ryo Suzuki, Lucas Figueredo, Loris D'Antoni, and Björn Hartmann. Writing Reusable Code Feedback at Scale with Mixed-Initiative Program Synthesis. In *Learning @ Scale*, 2017.

[21] Darren Key, Wen-Ding Li, and Kevin Ellis. I Speak, You Verify: Toward Trustworthy Neural Program Synthesis. *CoRR*, abs/2210.00848, 2022.

[22] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding Back-Translation at Scale. In *EMNLP*, 2018.

[23] Yewen Pu, Kevin Ellis, Marta Kryven, Josh Tenenbaum, and Armando Solar-Lezama. Program Synthesis with Pragmatic Communication. In *NeurIPS*, 2020.

[24] Hiroaki Funayama, Tasuku Sato, Yuichiroh Matsubayashi, Tomoya Mizumoto, Jun Suzuki, and Kentaro Inui. Balancing Cost and Quality: An Exploration of Human-in-the-Loop Frameworks for Automated Short Answer Scoring. In *AIED*, 2022.

[25] Rui Zhi, Samiha Marwan, Yihuan Dong, Nicholas Lytle, Thomas W. Price, and Tiffany Barnes. Toward Data-Driven Example Feedback for Novice Programming. In *EDM*, 2019.

[26] Ahana Ghosh, Sebastian Tschiatschek, Sam Devlin, and Adish Singla. Adaptive Scaffolding in Block-Based Programming via Synthesizing New Tasks as Pop Quizzes. In *AIED*, 2022.

[27] Anaïs Tack and Chris Piech. The AI Teacher Test: Measuring the Pedagogical Ability of Blender and GPT-3 in Educational Dialogues. In *EDM*, 2023.

[28] OpenAI. Codex-Edit. `https://beta.openai.com/playground?mode=edit&model=code-davinci-edit-001`, .

[29] OpenAI. Codex-Ccomplete. `https://beta.openai.com/playground?mode=complete&model=code-davinci-002`, .

[30] Zhiyu Fan, Xiang Gao, Abhik Roychoudhury, and Shin Hwei Tan. Automated Repair of Programs from Large Language Models. In *ICSE*, 2022.

[31] Georg Brandl, Matthäus Chajdas, and Jean Abou-Samra. Pygments. `https://pygments.org/`.

[32] Rohan Bavishi, Harshit Joshi, José Cambronero, Anna Fariha, Sumit Gulwani, Vu Le, Ivan Radicek, and Ashish Tiwari. Neurosymbolic Repair for Low-Code Formula Languages. *Proceedings ACM Programming Languages*, 6(OOPSLA2), 2022.

[33] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *ACL*, 2002.

[34] Tobias Kohn. The Error Behind The Message: Finding the Cause of Error Messages in Python. In *SIGCSE*, 2019.

[35] Ethan Caballero and Ilya Sutskever. Description2Code Dataset. `https://github.com/ethancaballero/description2code`, 2016.

[36] Yujia Li and et al. Competition-Level Code Generation with AlphaCode. 2022.

[37] Matthijs J Warrens. Five Ways to Look at Cohen's Kappa. *Journal of Psychology & Psychotherapy*, 5(4): 1, 2015.

[38] The Python Software Foundation. What's New In Python 3.10. `https://docs.python.org/3/whatsnew/3.10.html`, .

[39] The Python Software Foundation. What's New In Python 3.11. `https://docs.python.org/3/whatsnew/3.11.html`, .

[40] The Python Software Foundation. What's New In Python 3.12. `https://docs.python.org/3.12/whatsnew/3.12.html`, .

[41] William G Cochran. The $\chi 2$ Test of Goodness of Fit. *The Annals of Mathematical Statistics*, 1952.

[42] OpenAI. ChatGPT. `https://openai.com/blog/chatgpt`, 2023.

[43] OpenAI. GPT-4 Technical Report. *CoRR*, abs/2303.08774, 2023.

Modeling and Analyzing Scorer Preferences in Short-Answer Math Questions

Mengxue Zhang
UMass Amherst
mengxuezhang@umass.edu

Neil Heffernan
Worcester Polytechnic Institute
nth@wpi.edu

Andrew Lan
UMass Amherst
andrewlan@cs.umass.edu

ABSTRACT

Automated scoring of student responses to open-ended questions, including short-answer questions, has great potential to scale to a large number of responses. Recent approaches for automated scoring rely on supervised learning, i.e., training classifiers or fine-tuning language models on a small number of responses with human-provided score labels. However, since scoring is a subjective process, these human scores are noisy and can be highly variable, depending on the scorer. In this paper, we investigate a collection of models that account for the individual preferences and tendencies of each human scorer in the automated scoring task. We apply these models to a short-answer math response dataset where each response is scored (often differently) by multiple different human scorers. We conduct quantitative experiments to show that our scorer models lead to improved automated scoring accuracy. We also conduct quantitative experiments and case studies to analyze the individual preferences and tendencies of scorers. We found that scorers can be grouped into several obvious clusters, with each cluster having distinct features, and analyzed them in detail.

Keywords

Automated Scoring, Scorer Models, Bias

1. INTRODUCTION

Automated scoring (AS), i.e., using algorithms to automatically score student (textual) responses to open-ended questions, has significant potential to complement and scale up human scoring, especially with an ever-increasing number of students. AS algorithms are often driven by *supervised* machine learning-based algorithms and require a small number of example responses and their score labels to train on. These algorithms mostly consist of two components: a *representation* component that use either hand-crafted features [8, 17, 21, 27, 28, 37] or language models [24, 25, 34, 36, 42] to represent the (mostly textual) content in questions, student responses, and other information, e.g., rubrics [12] and a *scoring* component that use classifiers [4, 26] to predict the score of a response from its textual representation. In different subject domains, the representation component can be quite different, from hand-crafted features and neural language model-based textual embeddings in automated essay scoring (AES) [2, 27], automatic short answer grading (ASAG) [35, 47], and reading comprehension scoring [16] to specialized representations in responses where mathematical expressions are present [6, 31, 32, 40]. On the contrary, the scoring model does not vary significantly across different subject domains, often relying on simple classifiers such as logistic regression, support vector machines, random forests, or linear projection heads in neural networks [20]. We provide a more detailed discussion on related work in Section 1.2.

One key factor that limits the accuracy of AS methods is that the scoring task is a *subjective* one; human scorers are often given a set of rubrics [1] and asked to score responses according to them. However, different individuals interpret rubrics and student responses differently, leading to significant variation in their scores. For example, inter-scorer agreement can be as quite high in NAEP reading comprehension question scoring, with a quadratic weighted Kappa (QWK) score of 0.88 [16] and quite low in open-ended math question scoring, with a Kappa score of 0.083 (see Section 3.1 for details and Table 1 for a concrete example). This variation creates a *noisy labels* problem, which is a common problem in machine learning where one often needs to acquire a large number of labels via crowdsourcing [3, 18, 19]. In educational applications such as AS, this problem is even more important since the amount of labels we have access to is often small, which amplifies the negative impact of noisy score labels. Therefore, there is a significant need to analyze the preferences and tendencies of individual scorers, to not only improve AS accuracy by providing cleaner labels to train on but also understand where the variation in scores comes from and investigate whether we can reduce it.

1.1 Contributions

In this paper, we propose a collection of models for the variation in human scorers due to their individual preferences and tendencies, from simple models that use only a few parameters to account for the bias and variance of each scorer to complex models that use a different set of neural network parameters for each scorer. We ground our work in an AS task for short-answer mathematical questions and show that by adding our model to the classification component of

M. Zhang, N. Heffernan, and A. Lan. Modeling and analyzing scorer preferences in short-answer math questions. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 378–387, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115796

AS models, we can improve AS accuracy by more than 0.02 in Kappa score and 0.01 in AUC compared to AS methods that do not account for individual scorer differences. We also conduct qualitative experiments and case studies to analyze the individual preference and tendencies of scorers. We found that scorers can be grouped into several major, obvious clusters, with each cluster having distinct features, which we explain in detail. **We emphasize that our goal is NOT to develop the most accurate AS model; instead, our goal is to show that accounting for the variation across different individual scorers can potentially improve the accuracy of any AS model.**

1.2 Related work

Noisy labels. Individual scorers often exhibit different preferences and tendencies, as found in [38]. Some of our models for scorer preference and tendency are closely related to models used in peer grading [30], where students grade each others' work, which is often deployed in settings such as massive open online courses (MOOCs) where a large number of open-ended responses make it impossible for external human scorers to score all responses. Most of these models are inspired by methods in machine learning on combining labels from human labelers with different expertise in crowdsourcing contexts [41]. These models are simple and interpretable, with the most basic version involving a single bias parameter (towards certain score labels) and a single variance parameter (across different score labels) for each scorer. On the contrary, we experiment with not only these models but also more flexible but uninterpretable models, which are compatible with using pre-trained neural language models [13, 29] in the representation component of AS models.

AS and math AS. The majority of existing ASAG and AES methods focus on non-mathematical domains [7, 9, 11, 21, 27, 37, 39]. Recently, some AS methods are developed for specific domains that contain non-textual symbols, e.g., Chemistry, Computer Science, and Physics, which exist in student responses in addition to text, achieving higher and higher AS accuracy [5, 14, 23, 33, 34]. Our work is grounded in the short-answer math question scoring setting, which is studied in prior works [5, 6, 32, 46]. The key technical challenge here is that mathematical expressions that are often contained in open-ended student responses can be difficult to parse and understand in the representation component. The authors of [5] proposed a scoring approach for short-answer math questions using sentence-BERT (SBERT)-based representation of student responses and simply ignored mathematical expressions. The authors of [6] developed an additional set of features specifically designed for mathematical expressions and used them in conjunction with the SBERT representations as input to the scoring component. The authors of [32] fine-tuned a language model, BERT [13], further pre-trained on math textbooks, as the representation component; however, this representation was found to not be highly effective in later works [46]. The authors of [46] used a sophisticated in-context meta-training approach for automated scoring by inputting not only the response that needs to be scored but also scored examples to a language model, enabling the language model to learn from examples, which results in significant improvement in AS accuracy and especially generalizability to previously unseen questions.

Another line of related work is about fairness in educational data analysis since scorer preference can be classified as a form of individual bias. Researchers have proposed methods to incorporate constraints and regularization into predictive models to improve parity and mitigate fairness issues [10, 44, 45]. On the contrary, our work does not attempt at reducing biases; our focus is only on identifying a specific source of bias, individual scorer bias, in the AS context. Therefore, the only approach we use to mitigate biases is to leverage scorer identification information and investigate its impact on AS accuracy, following prior work on using this information in predictive models [43].

2. MODEL

We now detail our models for individual scorer preference and tendency in AS tasks. For all models, we use a BERT model [13] as the corresponding representation component of the AS model, which has been shown to perform well and reach state-of-the-art performance on the short math answer AS task with an appropriate input structure [46]. Let us denote each question-response pair that needs to be scored as q_i, while the j-th scorer assigns a score $y_{i,j} \in \{1, \ldots, C\}$ where C denotes the number of possible score categories.

2.1 Baseline

Our base AS model is one that directly uses the output `[CLS]` embedding of BERT as the representation of the question-response pair $\mathbf{r}_i \in \mathbb{R}^D$, where $D = 768$ is the dimension of the embedding. We also use a linear classification head (omitting the bias terms for simplicity) with softmax output [20] for all score categories, i.e.,

$$p(y_{i,j} = c) \propto e^{(\mathbf{w}_c^T \mathbf{r}_i) + b_c},$$

where $\mathbf{w}_c$ denotes the D-dimensional parameter for each score category and $b_c \in \mathbb{R}$ is the universal bias toward each score category.

2.2 Scalar bias and variance with scorer embeddings

The first version of our model is the simplest and most interpretable: we use a scalar temperature, i.e., variance parameter for each scorer, and a scalar offset, i.e., bias parameter on each score category for each scorer, i.e.,

$$p(y_{i,j} = c) \propto e^{\alpha_j (\mathbf{w}_c^T \mathbf{r}_i + b_{c,j})}, \tag{1}$$

where $\alpha_t > 0$ is the "temperature" parameter that controls the scorer's uncertainty across categories: larger values indicate higher concentrations of the probability mass around the most likely score category, which corresponds to more consistent scoring behavior. $b_{c,j} \in \mathbb{R}$ is the "offset" parameter that controls the scorer's bias towards each score category: larger values indicate a higher probability of selecting some score category, which corresponds to more positive/negative scoring preferences.

In practice, we found that parameterizing biases with a set of *scorer embeddings* lead to better performance than simply parameterizing the biases as learnable scalars. Specifically, we introduce a high-dimensional embedding for each scorer,

Table 1: Example questions, student responses, and scores. Some scorers assign highly different scores to similar responses.

question_id	*question_body*	*response*	*scorer_id*	*score*
43737	Chris spent $9 of the $12 he was given for his birthday. His sister Jessie says that he has spent exactly 0.75 of the money. Chris wonders if Jessie is correct. Explain your reasoning.	Jessie is correct because 0.75 in fraction form is 3/4. 9 is 3/4 of 12, so she is right.	1	4
		Jessie is wrong.	1	0
		she is correct	1	1
		Jessie is incorrect.	2	4
		Jessie is right because if you divide 12 by 9 you get 0.75.	2	2

$\mathbf{e}_j \in \mathbb{R}^D$, and use a $C \times D$ matrix $\mathbf{S}$ to map it to a low-dimensional vector that corresponds to the bias terms for all score categories. This advantage is likely due to the fact that more model parameters make the model more flexible and more capable in capturing detailed nuances in scorer preferences and tendencies.

2.3 Content-driven scorer bias and variance

In the models above, we have set the scorer biases and variances to be scorer-dependent but not question/response-dependent, i.e., the bias and variance of a scorer stay the same across all question-response pairs. However, in practice, it is possible that these parameters depend on the actual textual content of the question and the student's response. Therefore, we extend the scorer model in Eq. 1 into

$$\begin{aligned}&\mathbf{b}_{i,j} = f_b(\mathbf{r}_i, \mathbf{e}_j), \quad \alpha_t = f_\alpha(\mathbf{r}_i, \mathbf{e}_j),\\ &\text{where} \quad f_b(\mathbf{r}_i, \mathbf{e}_j) = \mathbf{r}_i^T \mathbf{A}_b \mathbf{e}_j, \quad f_\alpha(\mathbf{r}_i, \mathbf{e}_j) = \mathbf{r}_i^T \mathbf{A}_\alpha \mathbf{e}_j,\end{aligned}$$

where the bias $\mathbf{b}_{i,j}$ is now a $C \times 1$ vector of biases across all score categories and both question-response pair (i)-dependent and scorer (j)-dependent. f_b and f_α denote functions that map the textual representation of the question-response pair and the scorer embedding to the bias and variance parameters, which can be implemented in any way (from simple linear models to complex neural networks). In this work, we found that using bi-linear functions of the question-response pair representation $\mathbf{r}_i$ and the scorer embedding $\mathbf{e}_j$, using two $D \times D$ matrices $\mathbf{A}_b$ and $\mathbf{A}_\alpha$, results in the best AS accuracy.

2.4 Training with different losses

We explore using various different loss functions as objectives to train our AS model, which we detail below.

2.4.1 Cross-entropy

Since the AS task corresponds to a multi-category classification problem, the standard loss function that we minimize is the cross-entropy (CE) loss [20], summed over all question-response pairs and scorers, as

$$\mathcal{L}_{\text{CE}} = -\sum_{i,j} \sum_{c=1}^{C} \mathbf{1}_{y_{i,j}=c} \log p(y_{i,j} = c)$$

where $\mathbf{1}_{y_{i,j}=c}$ is the indicator function that is non-zero only if $y_{i,j} = c$. In other words, we are minimizing the negative log-likelihood of the actual score category among the category probabilities predicted by the AS model, $p(y_{i,j} = c)$.

2.4.2 Ordinal log loss

One obvious limitation of the standard CE loss is that it assumes that the categories are unordered, which works for many applications. Therefore, it penalizes all misclassifications equally. However, for AS, the score categories are naturally ordered, which means that score classification errors are not equal: if the actual score is 1 out of 5, then a misclassified score of 2 is better than 5, but they are weighted equally in the standard CE loss. Therefore, we follow the approach outlined in [15] and use an ordinal log loss (OLL), which we define as

$$\mathcal{L}_{\text{OLL}} = -\sum_{i,j} \sum_{c=1}^{C} |y_{i,j} - c| \log(1 - p(y_{i,j} = c)),$$

where we weight the misclassification likelihood, i.e., $-\log(1 - p(y_{i,j} = c))$, according to the difference between the actual score, $y_{i,j}$, and the predicted score, c. In the aforementioned example, this objective function would increase the penalty of a misclassified score of 5 by four times compared to a misclassified score of 2 when the actual score is 1, which effectively leverages the ordered nature of the score categories.

2.4.3 Mean squared error

Since the score categories are integers and can be treated as numerical values, one simple alternative to the CE loss is the mean squared error (MSE) loss, i.e.,

$$\mathcal{L}_{\text{MSE}} = \sum_{i,j} (y_{i,j} - \sum_{c=1}^{C} p(y_{i,j} = c)c)^2, \tag{2}$$

where we simply square the difference between the actual score and the expected (i.e., weighted average) score under the category probabilities predicted by the AS model.

3. QUANTITATIVE EXPERIMENTS

We now detail experiments that we conducted to validate the different scoring components of AS models and loss functions that capture scorer preferences and tendencies. Section 3.1 discusses details on the real-world student response dataset we use and the pre-processing steps. Section 3.2 details the evaluation metrics we use in our experiments. Section 3.3 details our experimental setting, and Section 3.4 details the experimental results and corresponding discussion.

3.1 Dataset

Table 2: Comparing different scorer models on short-answer math scoring. The combination of content-driven scorer bias and temperature with the OLL loss outperforms other scorer models and training losses.

Bias (b) & Temperature (α)	Loss Function	AUC	RMSE	Kappa
Universal (b_c, $\alpha = 1$)	CE	0.765 ± 0.003	0.954 ± 0.014	0.614 ± 0.009
Universal (b_c, $\alpha = 1$)	MSE	0.764 ± 0.003	0.946 ± 0.018	0.615 ± 0.008
Universal (b_c, $\alpha = 1$)	OLL	0.768 ± 0.003	0.944 ± 0.015	0.617 ± 0.006
Scorer-specific ($b_{c,j}$, α_j)	CE	0.768 ± 0.005	0.928 ± 0.023	0.628 ± 0.006
Scorer-specific ($b_{c,j}$, α_j)	MSE	0.772 ± 0.005	0.926 ± 0.025	0.625 ± 0.006
Scorer-specific ($b_{c,j}$, α_j)	OLL	0.770 ± 0.003	**0.916 ± 0.013**	0.628 ± 0.004
Content-driven ($b_{c,j}(\mathbf{r}_i)$, $\alpha_j(\mathbf{r}_i)$)	CE	0.772 ± 0.003	0.923 ± 0.016	0.631 ± 0.006
Content-driven ($b_{c,j}(\mathbf{r}_i)$, $\alpha_j(\mathbf{r}_i)$)	MSE	0.774 ± 0.004	0.922 ± 0.021	0.629 ± 0.005
Content-driven ($b_{c,j}(\mathbf{r}_i)$, $\alpha_j(\mathbf{r}_i)$)	OLL	**0.779 ± 0.004**	0.924 ± 0.013	**0.641 ± 0.005**

We use data collected from an online learning platform that has been used in prior work [5, 14], which contains student responses to open-ended, short-answer math questions, together with scores assigned by human scores. There are a total of 141,612 total student responses made by $25,069$ students to $2,042$ questions, with 891 different teachers being scorers. The set of possible score categories is from 0 (no credit) to 4 (full credit). The dataset mainly contains math word problems, where the answer could be mathematical such as numbers and equations or textual explanations, sometimes in the format of images.

We found that different scorers sometimes assign very different scores to the same response, which motivated this work. As an example, we analyze question-response pairs that are scored by more than one scorer and evaluate the Kappa score between these scorers. The *human* Kappa score is only 0.083, which means a minimal agreement between different scorers. Although there are only 523 such pairs, this case study still shows that even for the same exact response, scorers have highly different individual preferences and tendencies and may assign them highly different scores.

We also perform a series of pre-processing steps to the original dataset. For example, since some of the scorers do not score many responses, e.g., less than 100, there may not be enough information on these scorers for us to model their behavior. Therefore, we remove these scores from the dataset, which results in 203 scorers, $1,273$ questions, and $118,079$ responses. The average score is 3.152 ± 1.417. Table 1 shows some examples of data points of this dataset; each data point consists of the question statement, the student's response, the scorer's ID, and the score.

3.2 Metrics

We utilize three standard evaluation metrics for integer-valued scores that have been commonly used in the AS task [5, 14]. First, the area under the receiver operating characteristic curve (**AUC**) metric, which we adapt to the multi-category classification problem by averaging the AUC numbers over each possible score category and treating them as separate binary classification problems, following [22]. Second, we use the root mean squared error (**RMSE**) metric, which simply treats the integer-valued score categories as numbers, as detailed in Eq. 2. Third and most importantly, we use the multi-class Cohen's **Kappa** metric for ordered categories, which is often used to evaluate AS methods [1].

3.3 Experimental setting

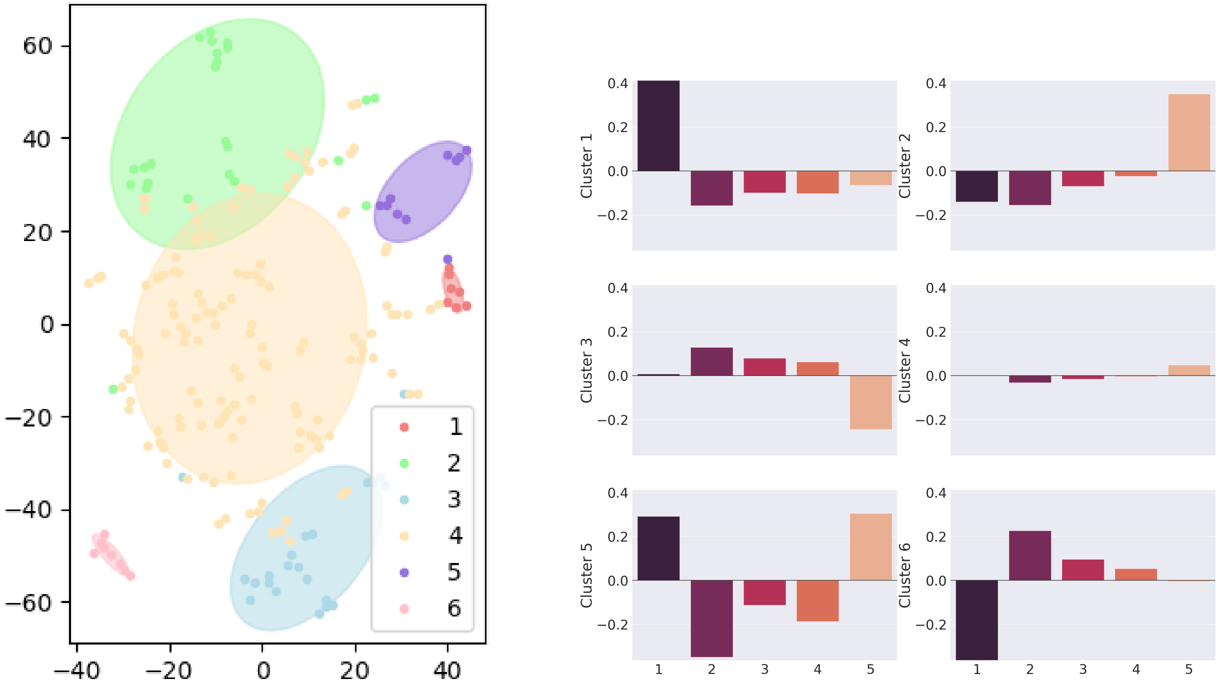
In the quantitative experiment, we focus on studying whether adding scorer information leads to improved AS accuracy. Therefore, when we are splitting a dataset into training, validation, and test sets, we ensure that every scorer is included in the training set. We divide the data points (question-response pairs, scorer ID, score) into 10 equally-sized folds for cross-validation. During training, we use 8 folds as training data, 1 fold for validation for model selection, and 1 fold for the final testing to evaluate the AS models.

For a fair comparison, every model uses BERT[1] as the pre-trained model for question-response pair representation, which has been shown to result in state-of-the-art AS accuracy in prior work [46]. We emphasize that our work on **scorer models** can be added on top of **any** AS method for response representation; applying these models on other AS methods is left for future work. We use the Adam optimizer, a batch size of 16, and a learning rate of $1e-5$ for 10 training epochs on an NVIDIA RTX8000 GPU. We do not perform any hyper-parameter tuning and simply use the default settings.

3.4 Results and discussion

Table 2 shows the mean and standard deviation of each scorer model trained under each loss function. We see that generally, models with content-driven scorer biases and variances outperform scorer-specific biases and variances, which outperform the base AS model that treats each scorer the same with universal values for bias and variance. The improvement in AS accuracy is significant, up to about 0.02 in the most important metric–Kappa, for the content-driven biases and variances over the standard AS approach of not using scorer information. This observation validates the need to account for individual scorer preferences and tendencies in the highly subjective AS task. Meanwhile, since the content-driven scorer bias and variance models outperform the scorer-specific bias and variance models, we can conclude that the content of the question and response does play an important role in scorer preference.

[1] `https://huggingface.co/bert-base-uncased`



(a) 2-D visualization of the learned scorer embedding space

(b) Bias for each score category

Figure 1: Visualization of clustering result on scorer embedding learnt via scorer-specific model. The left figure shows the 2-D visualization of scorer embedding space, and the right figure shows the average bias for each cluster

We also observe that training scorer models with the OLL loss outperform the other losse, while training with the MSE loss does not even lead to the best results on the RMSE metric. This observation suggests that taking into account the ordered nature of score categories instead of treating them as parallel ones is important to the AS task.

4. QUALITATIVE ANALYSIS

Despite the content-driven model delivering the highest AUC and Kappa results, the complexity of the information contained in its embedding space renders it difficult to interpret. Consequently, we have elected to concentrate on examining the scorer-specific model (detailed in Sec. 2.2).

4.1 Visualization of scorer embedding

Figure 1 shows a 2-D visualization of the learned scorer embedding space; We see that there are obvious clusters among all scorers. We then fit the learned scorer embeddings under a mixture-of-Gaussian model via the expectation-maximization (EM) algorithm with 6 clusters. The subfigures to each side of the main plot shows each cluster's average bias towards each score category, which are 0, 1, 2, 3, and 4 from left to right.

4.2 Features analysis based on each cluster

Cluster 1 shows a negative scoring profile, with a strong, positive bias towards the lowest score category 0 (positive $b_{c,j}$ values) and small, negative biases against higher scores, 1, 2, and 3 (negative $b_{c,j}$ values). These scorers assign 0 scores much more often than other score categories, compared to other scorers. The average score across question-response pairs is the lowest for this cluster, at 1.69. Meanwhile, this cluster has a relatively high score variance of 1.69, meaning that these scorers tend to have inconsistent behavior and assign a wide variety of score labels.

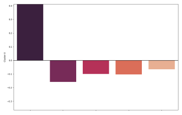
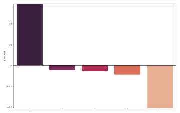
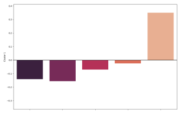
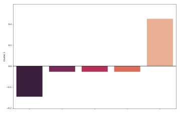
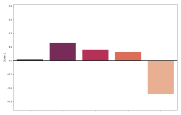
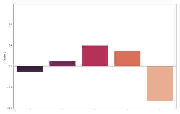
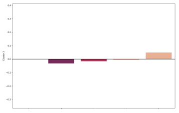
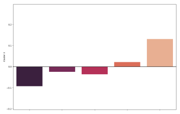
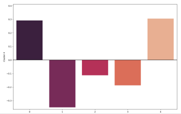
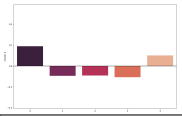
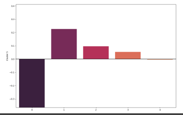
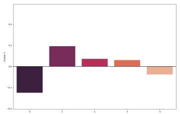
Cluster 2 shows a positive scoring profile, with a strong, positive bias towards the highest score, 4, and moderate negative biases against other scores. These scorers prefer to assign scores that are overwhelmingly higher compared to other scorers. The average score across question-response pairs is the lowest for this cluster, at 3.45. Meanwhile, this cluster has a relatively low score variance of 0.92, meaning that these scorers are consistent in scoring responses higher than other scorers.

Cluster 3 shows a conservative scoring profile, with small, positive biases towards the middling scores 1, 2, and 3 and a strong, negative bias against the top score 4. The average score across question-response pairs is 2.41 for this cluster with a variance of 1.4, which is high considering that scorers in this cluster rarely use the top score category, indicating that their scoring behavior is not highly consistent.

Cluster 4 shows an unbiased scoring profile, with a low bias towards or against any score category, with a slight preference for the top score category, 4. This cluster contains almost half of the scorers, which means that the majority of scorers are reliable (their scores depend mostly on the actual quality of the response, i.e., the $\mathbf{w}_c^T \mathbf{r}_i$ term in Eq. 1 rather than the bias term.

Cluster 5 shows a polarizing scoring profile, with strong, pos-

Table 3: Detailed biases and variance (inverse of temperature) for each scorer profile, their observed scoring distributions, and average response features. We normalize the observed scoring distributions to zero-mean, which makes them easier to visually compare against the learned biases. *math tok (%)* is the percentage of math tokens in the response. *img (%)* is the percentage of images in the response. *length* is the number of word tokens in the response.

Cluster	Bias	Observed scoring distribution (normalized)	Temperature	Score	Response features		
					math tok (%)	img (%)	length
1			1.013	1.685 ± 1.644	29.13	0.101	23.06
2			1.034	3.451 ± 0.919	32.12	1.286	24.40
3			0.996	2.415 ± 1.400	23.51	1.311	36.16
4			1.033	3.074 ± 0.991	29.48	0.304	21.94
5			1.026	2.558 ± 1.806	45.18	5.271	14.35
6			1.007	2.714 ± 1.331	33.83	1.403	13.34

itive biases toward both the lowest score, 0, and the highest score, 4, while having strong, negative biases against score categories in between. Scorers in this cluster often score a response as all or nothing while using the intermediate score values sparingly. The average score across question-response pairs is 2.55 for this cluster with a variance of 1.81, the highest among all clusters, which agrees with our observation that these scorers are highly polarizing and rarely judge any response to be partially correct.

Cluster 6 shows a lenient scoring profile, with a strong, negative bias against the lowest score, 0, and a moderate, positive bias towards the next score, 1, with minimal bias across higher score categories. Scorers in this cluster tend to award students a single point for an incorrect response instead of no points at all. The average score across question-response pairs is 2.71 for this cluster with a middling variance of 1.33.

5. CONCLUSIONS AND FUTURE WORK

In this paper, We created models to account for individual scorer preferences and tendencies in short-answer math response automated scoring. Our models differ from previous work by focusing on capturing the subjective nature of scoring rather than textual content. Our models range from simple to complex, with some using bias and variance as a function of the question and response. Our experiments on a dataset with low inter-rater agreement showed that accounting for scorer preferences and tendencies improved performance by more than 0.02 in the Kappa metric. Qualitative analysis showed obvious patterns among scorers, some with biases towards certain scores. Scorer-specific settings can model scorer grading behavior very well. In other words, the scorer's grading behavior is highly controllable, and the scorer's grading behavior representation is also well-represented in the hidden space. One practical extension could be adjusting the learned scorer bias by using a different type of scorer embedding to control model grading in a different scorer style. Future work can address limitations in our analysis. Our dataset only provides scorer IDs, lacking gender, race, or location. Investigating biases with this additional information is crucial, including how teacher-student relationships or shared demographics impact biases. Our analysis also did not consider student demographic information, which is important for fairness studies. Additionally, our scorer models were only validated with a BERT-based textual representation model, so further testing is needed to determine their adaptability to traditional, feature-based automated scoring methods.

6. ACKNOWLEDGEMENTS

The authors thank the NSF (under grants 1917713, 2118706, 2202506, 2215193) for partially supporting this work.

7. REFERENCES

[1] The ed.gov national assessment of educational progress (naep) automated scoring challenge. Online: https://github.com/NAEP-AS-Challenge/info, 2021.

[2] The hewlett foundation: Automated essay scoring. Online: https://www.kaggle.com/c/asap-aes, 2021.

[3] G. Algan and I. Ulusoy. Image classification with deep learning in the presence of noisy labels: A survey. *arXiv preprint arXiv:1912.05170*, 2019.

[4] Y. Attali and J. Burstein. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4:3, 2006.

[5] S. Baral, A. F. Botelho, J. A. Erickson, P. Benachamardi, and N. T. Heffernan. Improving automated scoring of student open responses in mathematics. *International Educational Data Mining Society*, 2021.

[6] S. Baral, K. Seetharaman, A. F. Botelho, A. Wang, G. Heineman, and N. T. Heffernan. Enhancing auto-scoring of student open responses in the presence of mathematical terms and expressions. In *International Conference on Artificial Intelligence in Education*, pages 685–690. Springer, 2022.

[7] S. Basu, C. Jacobs, and L. Vanderwende. Powergrading: a clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics*, 1:391–402, 2013.

[8] J. Burstein. The e-rater® scoring engine: Automated essay scoring with natural language processing. 2003.

[9] H. Chen and B. He. Automated essay scoring by maximizing human-machine agreement. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1741–1752, 2013.

[10] Y.-W. Chu, S. Hosseinalipour, E. Tenorio, L. Cruz, K. Douglas, A. Lan, and C. Brinton. Mitigating biases in student performance prediction via attention-based personalized federated learning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 3033–3042, 2022.

[11] A. Condor, M. Litster, and Z. Pardos. Automatic short answer grading with sbert on out-of-sample questions. *International Educational Data Mining Society*, 2021.

[12] A. Condor, Z. Pardos, and M. Linn. Representing scoring rubrics as graphs for automatic short answer grading. In *International Conference on Artificial Intelligence in Education*, pages 354–365. Springer, 2022.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[14] J. A. Erickson, A. F. Botelho, S. McAteer, A. Varatharaj, and N. T. Heffernan. The automated grading of student open responses in mathematics. In *Proceedings of the International Conference on Learning Analytics & Knowledge*, page 615–624, 2020.

[15] F. C. et al. A simple log-based loss function for ordinal text classification. online:https://openreview.net/pdf?id=khB9is39GvL, 2022.

[16] N. Fernandez, A. Ghosh, N. Liu, Z. Wang, B. Choffin, R. G. Baraniuk, and A. S. Lan. Automated scoring for reading comprehension via in-context bert tuning. In *International Conference on Artificial Intelligence in Education*, page 0, 2022.

[17] P. W. Foltz, D. Laham, and T. K. Landauer. The intelligent essay assessor: Applications to educational technology. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, 1(2):939–944, 1999.

[18] B. Frénay and M. Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013.

[19] Github. Awesome-learning-with-label-noise. https://github.com/subeeshvasu/Awesome-Learning-with-Label-Noise, 2020.

[20] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[21] A. C. Graesser, D. S. McNamara, M. M. Louwerse, and Z. Cai. Coh-metrix: Analysis of text on cohesion and language. *Behavior research methods, instruments, & computers*, 36(2):193–202, 2004.

[22] D. J. Hand and R. J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine learning*, 45(2):171–186, 2001.

[23] C. Leacock and M. Chodorow. C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 37(4):389–405, 2003.

[24] S. Lottridge, B. Godek, A. Jafari, and M. Patel. Comparing the robustness of deep learning and classical automated scoring approaches to gaming strategies. Technical report, Cambium Assessment Inc., 2021.

[25] E. Mayfield and A. W. Black. Should you fine-tune bert for automated essay scoring? In *15th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 151–162, 2020.

[26] D. S. McNamara, S. A. Crossley, R. D. Roscoe, L. K. Allen, and J. Dai. A hierarchical classification approach to automated essay scoring. *Assessing Writing*, 23:35–59, 2015.

[27] E. B. Page. The imminence of grading essays by computer. *The Phi Delta Kappan*, 47(5):238–243, 1966.

[28] I. Persing and V. Ng. Modeling prompt adherence in student essays. In *52nd Annual Meeting of the Association for Computational Linguistics*, pages 1534–1543, 2014.

[29] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[30] K. Raman and T. Joachims. Methods for ordinal peer grading. In *Proc. 20th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pages 1037–1046, Aug. 2014.

[31] A. Scarlatos and A. Lan. Tree-based representation and generation of natural and mathematical language. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023, preprint: https://arxiv.org/abs/2302.07974.

[32] J. T. Shen, M. Yamashita, E. Prihar, N. Heffernan, X. Wu, B. Graff, and D. Lee. Mathbert: A pre-trained language model for general nlp tasks in mathematics

education. *arXiv preprint arXiv:2106.07340*, 2021.

[33] S. Srikant and V. Aggarwal. A system to grade computer programming skills using machine learning. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1887–1896, 2014.

[34] K. Taghipour and H. T. Ng. A neural approach to automated essay scoring. In *Empirical methods in natural language processing*, pages 1882–1891, 2016.

[35] M. Uto and Y. Uchida. Automated short-answer grading using deep neural networks and item response theory. In *International Conference on Artificial Intelligence in Education*, pages 334–339, 2020.

[36] M. Uto, Y. Xie, and M. Ueno. Neural automated essay scoring incorporating handcrafted features. In *28th Conference on Computational Linguistics*, pages 6077–6088, 2020.

[37] S. Valenti, F. Neri, and A. Cucchiarelli. An overview of current research on automated essay grading. *Journal of Information Technology Education: Research*, 2(1):319–330, 2003.

[38] J. Z. Wang, A. S. Lan, P. J. Grimaldi, and R. G. Baraniuk. A latent factor model for instructor content preference analysis. *International Educational Data Mining Society*, 2017.

[39] Z. Wang, A. Lan, A. Waters, P. Grimaldi, and R. Baraniuk. A meta-learning augmented bidirectional transformer model for automatic short answer grading. In *Proc. 12th Int. Conf. Educ. Data Mining (EDM)*, pages 1–4, 2019.

[40] Z. Wang, M. Zhang, R. G. Baraniuk, and A. S. Lan. Scientific formula retrieval via tree embeddings. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 1493–1503. IEEE, 2021.

[41] J. Whitehill, T.-f. Wu, J. Bergsma, J. Movellan, and P. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in neural information processing systems*, 22, 2009.

[42] R. Yang, J. Cao, Z. Wen, Y. Wu, and X. He. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. *Findings of the Association for Computational Linguistics: EMNLP*, 2020:1560–1569, 2020.

[43] R. Yu, H. Lee, and R. F. Kizilcec. Should college dropout prediction models include protected attributes? In *8th ACM Conference on Learning@ Scale*, pages 91–100, 2021.

[44] M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *International Conference on World Wide Web*, pages 1171–1180, 2017.

[45] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. Learning fair representations. In *International Conference on Machine Learning*, pages 325–333, 2013.

[46] M. Zhang, S. Baral, N. Heffernan, and A. Lan. Automatic short math answer grading via in-context meta-learning. *arXiv preprint arXiv:2205.15219*, 2022.

[47] Y. Zhang, R. Shah, and M. Chi. Deep learning+ student modeling+ clustering: A recipe for effective automatic short answer grading. In *International Conference on Educational Data Mining*, page 562, 2016.

APPENDIX

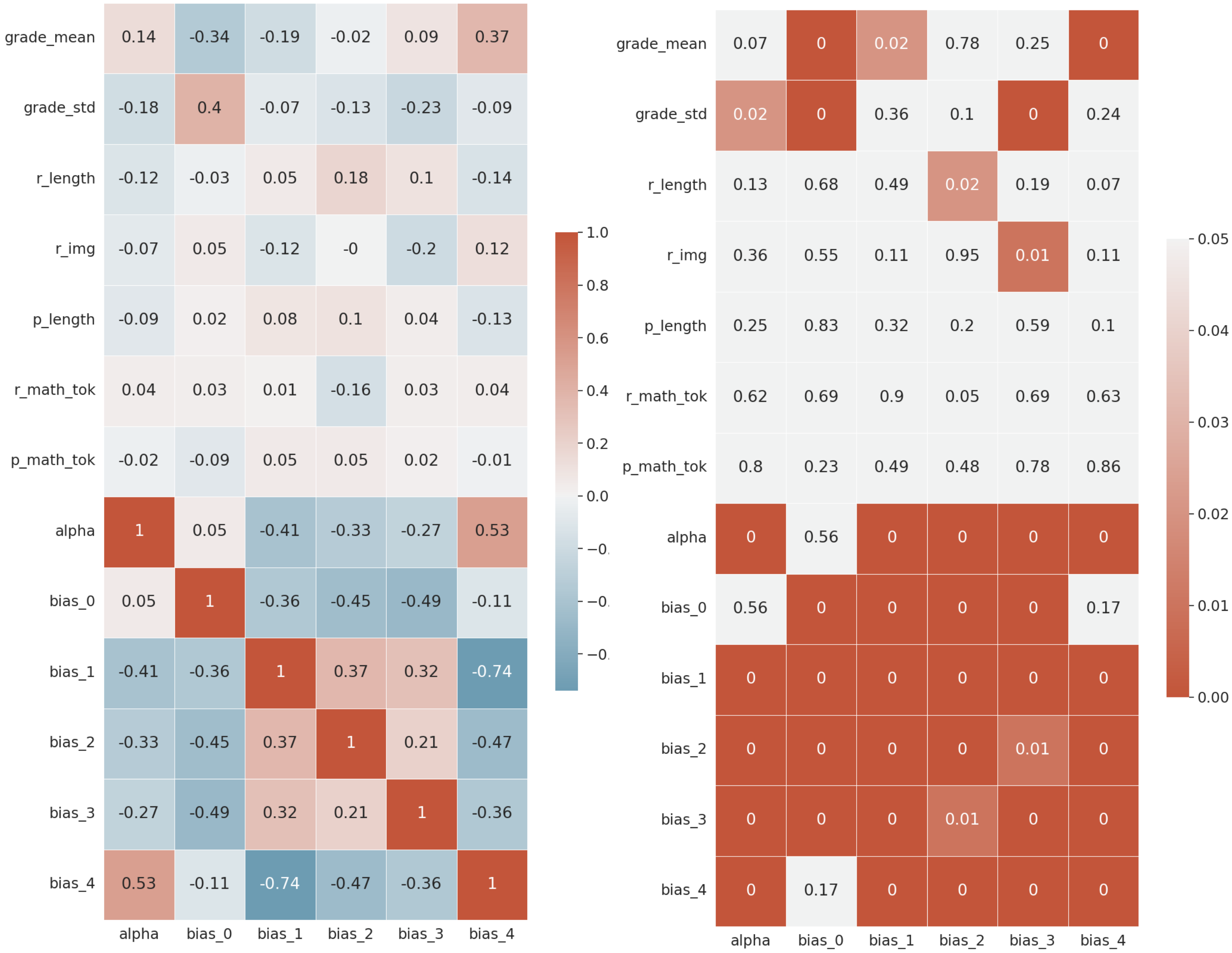
A. CORRELATION ANALYSIS

In Table 3, we see that the learned scorer biases for each cluster are highly correlated with the observed score distribution across score categories. However, it is not obvious how the variance, i.e., the inverse of the temperature parameter (α), correlates with other model parameters and response features. Therefore, we calculate the correlation coefficient (left) and the corresponding p-value (right) between each pair of model parameters and response features and show them in Figure 2. In the left part of the figure, we see that α positively correlates with the mean of scores and negatively correlates with the standard deviation of scores. In the right part of the figure, we see that α is significantly correlated with the standard deviation of scores, which is expected since this temperature parameter is designed to capture the variation in score category assignments. We also see that α is also significantly correlated with the bias terms of each score category, with a positive correlation with the bias for score category 4 and negative correlation with the bias for other categories.

For the bias terms, we see that most of the biases are significantly correlated with the mean and standard deviation of scores, but less correlated with question-response pair features. This observation suggests that the bias terms mainly depend on scorer behavior rather than the question-response pair, which is what the model intended to do; the question-response pair is captured by the $\mathbf{w}_c^T \mathbf{r}_i$ term in Eq. 1. The bias for score category 2, however, does not significantly correlate with the mean and standard deviation of scores but significantly correlates with other question-response pair features. One possible explanation is that since this score category is the middle of all scores, scorers do not show any bias towards or against this score category and can solely rely on the actual content of the question and response. , for example, the length of the response which might show that bias 2 does not accurately represent scorer grading behavior.

B. CASE STUDY: SAME SCORER, DIFFERENT RESPONSES

Table 4 shows several examples of different questions and responses and corresponding scores for a single scorer, with the actual score, biases calculated from the content-driven scorer bias and variance model, and predicted scores for different models. The overall bias for this scorer is $[-0.043, -0.36, -0.212, 0.061, 0.439]$ across all score categories, which indicates that this scorer prefers to assign high scores (especially the full score 4) but often assigns low scores except the lowest score (0). Overall, we see that if we do not include biases in the AS model (the sixth column), the AS model tends to predict middling scores, while the human scorer tends to give students full credit (4). For Question 2, this example shows that the content-driven scorer bias model captures nuanced scorer preference: for the meaningless response “idk”, which should have a score of 0, the scorer has a strong preference towards giving it a high score (3). This bias only appears for seemingly meaningless responses but not overall (overall bias towards score category 3 is minimal at 0.061). Therefore, we see that the scorer-specific model cannot capture this information since its biases and variance are global across all question-response pairs for this scorer. As a result, content-driven scorer models are more flexible in handling these cases compared to other models, which is also evident in the quantitative results in Table 2 that this model achieves the highest overall AS accuracy.



(a) Correlation Coefficient Matrix

(b) P-values Matrix

Figure 2: Correlation coefficients and corresponding p-values across the bias, variance terms, and response features for the scorer-specific bias and variance models.

Question id	Response	Actual score	Content-driven prediction	Scorer-specific prediction	No bias prediction	Content-driven scorer bias
1	The graph was touching the origin, but it didn't have a straight line	4	4	4	3	[-0.61, -1.29, 0.04, -0.33, 1.33]
2	It meets the origin and it goes perfectly diagonal.	3	4	4	3	[-0.26, -1.80, -0.57, -0.39, 2.09]
	Because it's a straight line that goes through the orgin	4	4	4	3	[0.13, -1.53, -0.76, -0.47, 1.71]
	its porportional because it has a straight line and and starts at the bottom.	3	3	4	2	[1.19, -0.20, -3.18, 1.24, 1.21]
	idk	3	3	0	0	[-6.26, -0.09, 2.76, 4.56, 1.50]

Table 4: Examples of student response and scores for a single scorer with biases $-0.043, -0.36, 0.061, -0.212, 0.439$ for all score categories. Notice that the no-bias prediction is the prediction of the content-driven model that does not scale with bias.

Mining Detailed Course Transaction Records for Semantic Information

Yinuo Xu
Department of Statistics
University of California, Berkeley
yinuoxu54@berkeley.edu

Zachary A. Pardos
Berkeley School of Education
University of California, Berkeley
pardos@berkeley.edu

ABSTRACT

In studies that generate course recommendations based on similarity, the typical enrollment data used for model training consists only of one record per student-course pair. In this study, we explore and quantify the additional signal present in course transaction data, which includes a more granular account of student administrative interactions with a course, such as wait-listing, enrolling, and dropping. We explore whether the additional non-enrollment records and the transaction data's chronological order play a role in providing more signal. We train skip-gram, FastText, and RoBERTa models on transaction data from five years of course taking histories. We find that the models gain moderate improvements from the extra non-enrollment records, while the chronological order of the transaction data improves the performance of RoBERTa only. The generated embeddings can also predict course features (i.e. the department, its usefulness in satisfying requirements, and whether the course is STEM) with high accuracy. Lastly, we discuss future work on the use of transaction data to predict student characteristics and train course recommender models for degree requirements.

Keywords

Higher education, course recommendation, course2vec, enrollment records, chronology

1. INTRODUCTION

Prior work has shown that standard enrollment data can be used to infer content similarities between courses within [11] and across institutions [10]. One hypothesis for how these models capture semantics, is that they encode students' aggregate perceptions of a courses as communicated by the contexts in which they are selected by a student (e.g., in which semester and along with which other courses) [13]. Transaction data includes additional student actions and, therefore, more potentially inferable student perceptions of course semantic information and course similarity.

Two important features of transaction data are: 1) it contains more granular information on top of student enrollment actions, such as waitlisting and dropping, including students' reasons for doing so; 2) the order of the student actions is available as these granular actions are timestamped instead of just term-stamped, as is the case with conventional enrollment data that has inhibited chronological sorting within semester in past work. We hypothesize that both features might provide more granular semantic information and similarity signal, and thus improve similarity-based recommendations. In this study, we first present summary statistics of transaction data and features of courses that will be used in subsequent analyses. Next, we present the methodology and results related to the following research questions:

- **RQ1**: Does the extra non-enrollment transactions (waitlisting and dropping) provide additional course similarity signal on top of enrollment transactions?
- **RQ2**: Does the chronological order of the transaction data provide more course similarity signal than random (within-semester) order?

We quantify the amount of signal gained by these two features by comparing the performance of skip-gram, FastText, and RoBERTa models with varying additional non-enrollment records and orders. We further investigate how well the model representations from transaction and enrollment-only data represent other semantic features of courses by predicting features of the courses, such as department, STEM /non-STEM designation, student major diversity, utility in fulfilling requirements, and course popularity.

We find that the extra non-enrollment records do provide more similarity signal. Most models trained with full transaction data perform better than the baseline models that are only trained on enrollment actions. We also find that the chronological order of the transaction data does not improve the course similarity signal for skip-gram and FastText, but does improve the signal for RoBERTa. Lastly, we find that the best embedding model (FastText trained on transaction data) is able to predict course features better than the best embedding model trained only on enrollment actions.

2. RELATED WORK

Y. Xu and Z. A. Pardos. Mining detailed course transaction records for semantic information. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 388–395, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115794

Big Data is one of the driving forces behind educational recommender systems and learning analytics as there is an increasing volume, variety, and integrity of data obtained from various educational platforms [2]. Furthermore, as the use of MOOCs (massive open online courses) and other digital platforms has increased, student data (ranging from student enrollment data to behavioral data like clickstream) has become more granular. Representations of students, lessons, and assessments from historical lesson student-content interactions in an online tutoring system are used to create personalized lesson sequence recommendations [14]. Students' daily activities, including potentially sensitive, private data, could be used to predict their success in online courses using supervised machine learning systems [3].

In traditional higher education institutions, there have been efforts to conduct predictive modeling and create course recommender systems using a variety of novel institutional data. These data include enrollment histories, major declarations, and catalog descriptions. Large-scale syllabus data was introduced as a novel source of information on tasks of predicting course prerequisites, credit equivalencies, student next semester enrollments, and student course grades [6]. It was found that course descriptions resulted in the highest signal representation accuracy in predicting course similarity, the prediction task we are also concerned with. Student enrollment data and course catalog data were also used to create course recommendations given students' academic interests and backgrounds at a liberal arts program [9]. The course recommendations were based on topic modeling on course catalog descriptions, and were found to be relevant for a wide range of academic interests. A course recommendation system based on score predictions with cross-user-domain collaborative filtering was developed using course-score records based on different student majors [4]. In particular, the algorithm was designed to effectively predict the score of the course for each student by using the course-score distribution of the most similar senior students.

More recently, it has been shown that incorporating data from multiple heterogeneous sources improves course recommendations [5, 16]. Specifically, sources such as course, student, and career information are integrated with ontology-based personalized course recommendation to help students gain comprehensive knowledge of courses based on their relevance. Course description data was integrated with job advertisement data to identify necessary job skills [17] with a hybrid course recommendation system to extract relevant skills and entities, and provide recommendations on multiple individualized levels of university courses, career paths with job listings, and industry-required with suitable online courses. Course description and job advertisement data were also used to build a heterogeneous graph approach for cross-domain recommendation for both students and professionals[19]. However, student enrollment data used in these previous studies only have contained courses that students enroll in each semester, as the type of granular and detailed student behavioral data are not always readily available in traditional formal higher education learning environments. Our study utilizes a new source of detailed, more granular course enrollment data for course similarity-based recommendation.

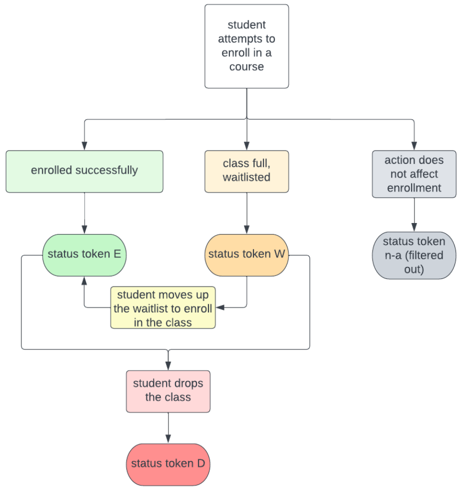

student attempts to enroll in a course
enrolled successfully
class full, waitlisted
action does not affect enrollment
status token E
status token W
status token n-a (filtered out)
student moves up the waitlist to enroll in the class
student drops the class
status token D

Figure 1: Transition diagram for enrollment status

Nascent findings [18] on the application of FastText to course equivalency task, found that there is 97.95% improvement in model performance from skip-gram to FastText. Since course names are morphologically rich, usually with information such as the department, level of division, and whether it is cross-listed, we expect that transaction data would provide more course similarity signal. Additionally, transaction data contains non-enrollment tokens that we concatenate to the end of course names (i.e. English 100_W denotes a course that is waitlisted) to distinguish various actions. We take advantage of such additional tokens with subword representations from FastText and RoBERTa.

3. DATA

The transaction data was provided through official channels at UC Berkeley, a large public university in the US. It shows a history of students enrolling, dropping, and waitlisting into classes, with each row representing one of these actions with a specific timestamp It is set to be from 2016 Fall — 2022 Summer. Table 1 contains the size and number of unique courses of the original transaction data, and its various filtered versions. Table 2 shows an example of the transaction data where one student (xxxxxx123) enrolls in 110 Math and gets waitlisted in 150 Molecular & Cell Biology; while another student (xxxxxx456) attempts to drop 148 Sociology but is unsuccessful. Figure 1 shows how enrollment status token changes based on user actions. Each action would generate a row of records with the corresponding status token. For example, when a student attempts to enroll in a course, there are 3 scenarios: 1) enroll successfully and their action is recorded with the status token ”E”; 2) waitlist in the class and their action is recorded with the status token ”W”; or 3) their action does not affect their enrollment status and it is recorded with the status token ”n-a” which we filtered out. At a later time, when the student moves up the waitlist and successfully enrolls in the class, another record with token ”E” will be recorded. And any time when a student drops a course, status token ”D” will be recorded.

Table 1: Size and number of unique courses of transaction data and its various filtered versions

Data	size	unique courses
Transaction original	11,136,719	16,686
Transaction filtered (student initiated, action affects status)	9,141,091	9,251
Transaction filtered (student initiated, action affects status, outcome status = E)	2,807,265	8,817
Transaction filtered (student initiated, action affects status, outcome status = E, D)	4,335,464	9,025
Transaction filtered (student initiated, action affects status, outcome status = E, D, W)	5,273,907	9,033

Table 2: Example of transaction data

Student id	Enrollment request timestamp	Semester	Course	Enrollment status outcome	Enrollment message
xxxxxx123	2021-10-11 15:22:15	Fall 2021	150 Molecular & Cell Biology	W	You have been placed on the waitlist in position number 3.
xxxxxx123	2021-10-11 15:29:17	Fall 2021	110 Math	E	Your enrollment request has been processed successfully.
xxxxxx456	2021-3-11 20:48:39	Spring 2021	148 Sociology	NaN	You cannot drop this class.

While the transaction data contains various features associated with student enrollment actions such as the source and reasons for the enrollment request, we focus on ”enrollment status outcome”. It contains token “D” (dropped), “E” (enrolled), “W” (wait-listed), or n-a (when the action does not affect enrollment), and the enrollment message contains the description accompanying the enrollment status outcome. The top 3 enrollment status messages corresponding to the four status tokens are listed below:

- Token “D”
  - A Grade of [LETTER] has been assigned for this Drop Request.
  - Your enrollment request has been processed successfully.
  - Warning - Enrollment status is Withdrawn.
- Token “E”
  - Your enrollment request has been processed successfully.
  - You have already taken this class.
  - Invalid Access to Override Class Links
- Token “W”
  - You have been placed on the waitlist of [CLASS] in position number [NUMBER].
  - The Requirement Designation Option was set to ’YES’ by the enrollment process.
  - Course previously taken and may be subject to institutional repeat policy.
- Token “n-a”
  - Unit Limit Exceeded For Appointment Period.
  - You are unable to enroll in this class at this time.
  - Class [CLASS] is full.

Table 3: Summary table of statistics on the transaction data

	Median number of actions	Top 3 departments	Proportion of STEM/ non-STEM	Proportion of different divisions	Median course requirement lists satisfied
Enrolled records	2,408.5	Computer Science, Business Admin-Undergrad, Statistics	54% STEM, 56% non-STEM	59% upper, 41% lower	18
Waitlisted records	6,188	Computer Science, Mathematics, Chemistry	66% STEM, 34% non-STEM	58% lower, 42% upper	12.5
Dropped records	6,188	Computer Science, Business Admin-Undergrad, Mathematics	58% STEM, 42% non-STEM	55% lower, 45% upper	16

We then filtered the transaction data to only contain student-initiated actions and actions that affect enrollment status, filtering out the records with the enrollment status token ”n-a” and removing 20% of the rows. Actions that are not student-initiated include those that are initiated by the administration (e.g., manual enrollment of a student), and those that are batch-processed (e.g., if a class is canceled, then everyone is dropped). Actions that affect enrollment status include dropping, enrolling, enrolling from waitlist, or dropping to waitlist. Actions that do not affect enrollment status include adding grade, changing grade, or changing waitlist position.

3.1 Analysis of Transaction Data

We conducted data analysis of the transaction data to explore the types of courses students get enrolled, waitlisted for, and dropped the most. The categories we investigate are: whether the course is STEM, its divisions, and its usefulness in satisfying degree requirements. We select the top 100 courses to analyze. As show in Table 3, the courses for all three actions include popular courses in Computer Science and Business Administration. The majority of waitlisted and dropped records contain STEM courses and lower division courses, while the majority of enrolled records contain non-STEM and upper division courses. Enrolled records contain the highest median number of course requirement lists that are satisfied, followed by dropped, and then waitlisted records. We further investigate these features in a later section to quantify the prediction power of course embeddings trained on the transaction data.

4. MODELS

We apply three embedding models to the transaction data. Two of the models, skip-gram and FastText, have been evaluated on a similar task of course equivalencies. The third model is a SentenceTransformer network architecture with a custom trained RoBERTa model as the word embedding model layer.

4.1 skip-gram Course2Vec

The Course2Vec model, like a Word2Vec model, learns course representations by treating an enrollment sequence as a sentence and each class in the sequence as a word [12, 11]. To distinguish between courses associated with different enrollment outcomes, the enrollment status outcome token is concatenated to the end of the course (e.g. ”Math 10A_E”) before passing the course sequence to the transaction model. A transaction sequence with the token concatenated is [”Molecular & Cell Biology 160_E”,”Molecular & Cell Biology 160_D”, ”Statistics 134_W”], in which a student enrolls in Molecular & Cell Biology 160, drops it, and is waitlisted in Statistics 134.

4.2 FastText Course2Vec

The department name, course affixes, and course number are typically included in course titles, which have a rich morphological structure. Prefixes like ”C” in ”History C140” indicate a course that is taught jointly by two or more departments, whereas suffixes like ”A” or ”B” in ”Chemistry 1A” and ”Chemistry 1B” indicate courses that should be taken sequentially. At UC Berkeley, lower-division courses, upper-division courses, and graduate-level courses are designated by course numbers below 100, 100-199, and 200 and above, respectively. So FastText [1], which represents words as a bag of character n-grams and is able to compute out-of-vocabulary words, is expected to take advantage of the extra enrollment status tokens of the transaction data.

4.3 Sentence Transformer with RoBERTa

RoBERTa [8] is a modification of the original BERT model that is trained on a much larger dataset and removes the Next Sentence Prediction objective. We first trained a byte-level Byte-pair encoding tokenizer rather than a WordPiece tokenizer like BERT to make sure all words will be decomposable into tokens as it builds its vocabulary from single bytes. Each transaction sequence is again treated like a sentence. We then trained the RoBERTa model from scratch on a task of Masked Language Modeling, noting the similarity in completing a sentence and suggesting a course sequence. Next, we constructed a Sentence Transformer network [15] using the RoBERTa word embedding model and a mean pooling layer. For RoBERTa models trained on multiple enrollment status tokens, we derived the course embeddings using a sentence consisting of all the course's related tokens (i.e. Data 100_E, Data 100_D).

4.4 Model Training and Evaluation

We use the equivalency validation set containing 480 course credit equivalency pairs maintained by the Office of Registrar as ground truth for course similarity. To increase the validation set, we swap the pairs, resulting in 960 pairs. We then filtered the validation set to include only courses that occur in the intersection of all filtered data (8,817 unique courses as shown in table 1) and courses that could be predicted by all models, yielding a total of 784 pairs.

Recall@10 is calculated for equivalency validation pairs using the model evaluation metrics and validation dataset (containing pairs of courses with equivalent credits) established in a previous study [12]. We find similar courses to the first course for each validation course pair by ranking other courses based on cosine similarity of their vector representations, and we calculate recall@10 based on the rank of the second course. For transaction Course2Vec models that take into account enrollment status tokens – for instance Course2Vec trained on classes with enrollment token “W” – equivalency pairs that could not be predicted because either one does not exist in the vocabulary set of Course2Vec (token “W”) are then predicted by a Course2Vec model trained on non-token sequences. To obtain a single embedding for a course with a Course2Vec model trained on multiple versions of the course with different enrollment tokens (E, D, W), we use 3 different methods: 1) simply use this model to get the embedding for the original course without any tokens (i.e. Math 1A), 2) average and 3) concatenate the embeddings of the various versions of the course with different tokens. To obtain a course embedding with Sentence Transformer based on a RoBERTa trained on transaction records with different tokens, we use 2 methods: 1) simply pass in the course without any tokens; 2) pass in a synthetic course sequence containing various versions of the course with different tokens (i.e. Math 1A_E, Math 1A_D).

We then use ten-fold cross-validation to select the best model hyper-parameters. We split the 784 validation pairs into 10 folds. Then, within each phase of the cross-validation, 80% of the validation pairs are used to find the best training hyper-parameters, which are then used to create a model to evaluate on the rest of the 20% of the pairs. The ranks of the test pairs are recorded for each fold, then they are appended together to calculate overall recall@10. We don't use temporal cross-validation because the validation set consists of similarity pairs that do not have established dates associated with them. Grid search is used on the following hyperparameter space for both skip-gram and FastText:

- Min count: [10, 20, 30, 40, 50, 60, 70, 80, 90]
- Window: [2, 3, 4, 5, 6, 7, 8, 9]
- Vector size: [200, 210, 220, 230, 240, 250, 260, 270, 280, 290, 300, 310]
- Sample: [3.e-05, 2.e-05, 1.e-05]
- Alpha: [0.01, 0.02, 0.03, 0.04]
- Min alpha: [0.0001, 0.0003, 0.0005, 0.0007]
- Negative: [10, 15, 20, 25]

See Appendix A.1 for optimal hyperparameters for the best model. The optimal hyperparameters for our data are likely to differ from others' based on size of course catalog and number of enrollments.

5. RESULTS

5.1 RQ1: Utility of non-enrollment transactions records

We found that transaction records do provide more course similarity signal. As shown in table 4, most models trained with enrollment (E) and non-enrollment (D& W) transactions (whether the tokens are hidden, averaged, concatenated) show improvements from the baseline models that

Table 4: Percent improvement of the best models trained on transactions (E,D,W) from baseline models trained only on enrollment (E)

	Baseline (random) recall	Random	Baseline (ordered) recall	Ordered
Skipgram	0.296	2.53% (E&D/ no token)	0.244	0% (E&D/ no token)
FastText	0.446	4.22% (E&D/ avg)	0.367	14.5% (E&D&W/ concat)
RoBERTa	0.309	4.96% (E&D&W/ no token)	0.347	-2.19% (E&D&W/ no token)

Table 5: Percent improvement of the best models trained on chronologically ordered transaction records from those trained on randomly ordered records

	E	E & D	E & W	E & D & W
Skipgram	0.296-17.6%	0.304-19.6%	0.296-22.6%	0.295-17.9%
FastText	0.446-17.6%	0.464-11.9%	0.418-11.9%	0.458-7.67%
RoBERTa	0.309+12.4%	0.320+3.19%	0.305-3.35%	0.324+4.72%

are only trained on enrollment records. Only skip-gram and RoBERTa trained on chronologically ordered transaction records do not show any improvement from their respective baseline models trained on chronologically ordered enrollments. Additionally, we see that the best models that outperform the baseline E model are either models trained with the tokens E&D, or models trained with E&D&W. The enrollment status token W does not improve from the baseline model, as models trained on E&W perform worse than the baseline E models, suggesting that W transactions could be random noise. The greatest percent improvement from enrollment to transaction records is to use FastText trained on the full records (E,D,W) and ordered sequences. Overall, the best-performing model for skip-gram is that trained on E&D random order (no token), the best FastText model is that trained on E&D random order (average token), and the best RoBERTa model is that trained on E chronological order (no token).

5.2 RQ2: Utility of chronological transactions records

The chronological order of the transaction data does not improve the course similarity signal for skip-gram and FastText, but does improve the signal for RoBERTa, as shown in table 5. The greatest percent decrease in performance from randomization to chronology is skip-gram trained on E & W transactions, and the greatest percent increase in performance from randomization to chronology is RoBERTa trained only on enrollments.

In general, the best model overall is FastText trained on random (within-semester) ordered E & D transactions (evaluated by averaging the E and D embeddings), with a recall@10 of 0.464. And the best model trained only on enrollment events is FastText trained on randomly (within-semester) ordered events, with a recall of 0.445.

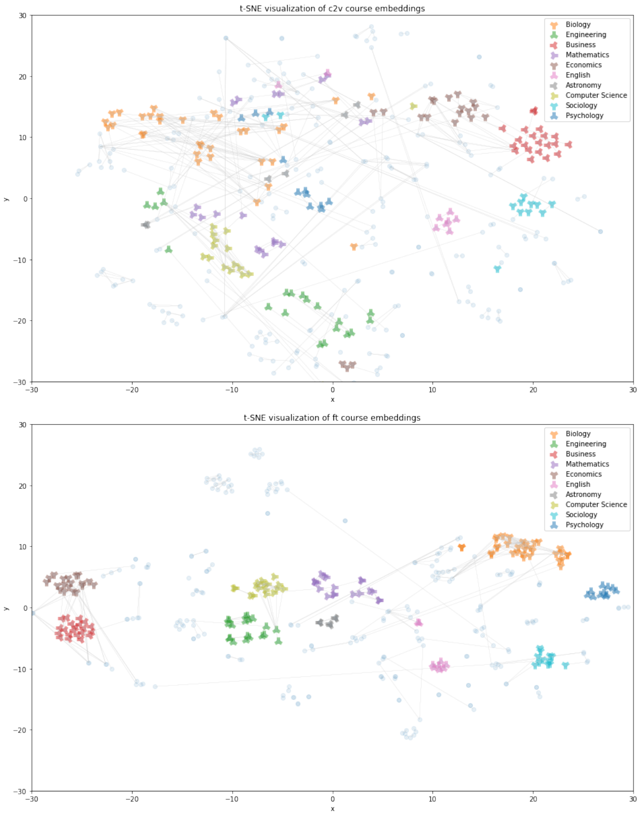


Figure 2: TSNE visualizations of courses in selected departments created by skip-gram and FastText

5.3 Visualizing embeddings

To provide an intuitive explanation for the increase in recall@10 from course2vec skipgram to FastText, we present comparisons of the TSNE visualizations of courses in randomly selected departments produced by these 2 models (Fig.2.). We chose to not present all departments to avoid overcrowding the visuals. The colored points indicate different departments, the transparent blue points indicate the rest of the courses in the validation pairs, and the faint grey lines indicate connections between equivalency pairs. Visually, we see that the FastText embeddings appear more closely clustered than the Course2Vec skipgram embeddings.

6. ANALYZING PREDICTIVE POWER

We investigate how well we could predict the various features of the courses using the best model trained on extra non-enrollment actions vs. the best model trained on only enrollment actions. These features include whether the course is STEM (binary), the department of the course (80 categories), the division (3 categories), diversity of student majors enrolled in the course (binary), the course's utility to satisfy requirements (binary), and its popularity (binary). Diversity (the number of different types of unique student majors enrolled in the course), requirement utility (the number of requirement lists the course satisfy), and popularity (the frequency of student interactions with the course) are made into binary variables by categorizing the course as be-

Table 6: Accuracy of baseline majority, logistic regression, and MLP in predicting course features, using the baseline enrollment (E) and best transaction (E,D) embeddings

	Baseline majority	Logistic regression (E)	MLP (E)	Logistic regression (E, D)	MLP (E,D)
STEM/ non-STEM	0.519	0.993	0.994	0.995	0.999
department	0.0885	0.998	0.987	0.995	0.998
division	0.549	0.994	0.984	0.984	0.998
student major diversity	0.785	0.816	0.975	0.944	0.983
course requirement utility	0.510	0.816	0.950	0.950	0.989
popularity	0.472	0.846	0.963	0.914	0.984

low and equal to or above the median value. We compared the accuracies (Table 6) of a baseline majority, logistic regression, and MLP classifier using the best embeddings of the enrollment actions (FastText trained on randomly ordered E records) and the best embeddings of transaction actions (average FastText embedding trained on randomly ordered E & D records), obtained through 5-fold cross validation. See Appendix A.2 for the optimal hyperparameters for logistic regression and MLP.

In general, for all models, the best transaction embedding is able to improve on the enrollment embedding. For both embeddings, logistic regression and MLP models are able to achieve almost perfect accuracy on predicting STEM/non-STEM, department, and division of the courses in the validation pairs. The biggest improvement comes from course requirement utility (16.4% increase for logistic regression). Overall, transaction embeddings have great predictive power in classifying various course features.

7. DISCUSSION & FUTURE WORK

Does chronology add more similarity signals to enrollment data? Our results suggest that there is no more signal in chronology than randomization. Overall, the best-performing model for skip-gram is that trained on E&D random order, the best FastText model is that trained on E&D random order, and the best RoBERTa model is that trained on E chronological order. This suggests that these models are more likely to pick up on course similarity signals when the data contains transactions (E,D,W) and are randomly ordered. The reason that randomized course sequences work better than ordered ones could be that randomization gives courses more contexts, especially popular courses. Popular courses are more likely to be chosen first in a course sequence for a semester, meaning that they may have fewer different courses in their context window than other courses during training for skip-gram and FastTexts, compared to courses that are chosen in the middle of the sequence. However, for chronologically ordered transactions, FastText is the only model that's able to pick up more signal, likely because of its ability to take advantage of the morphological structure of course names, despite the potential negative effect of chronology in reducing contexts. Future work could focus on investigating further the reason why randomization provides better similarity signal than chronology.

There are several other areas of additional future work. First, a limitation of our work is that it may not be practical for many institutions to collect or utilize transaction data. These data are rare, so we only had one institution's dataset to analyze, limiting our ability to make claims on generalizability. Future work could focus on investigating whether the same conclusions hold for transaction data of other institutions. Second, the fact that RoBERTa is able to benefit from the signal of the chronology of the transaction data, while FastText benefits from the random order could justify future work into combining the embeddings of FastText and RoBERTa. Next, we could explore better ways of obtaining course embeddings from RoBERTa to take advantage of its contextual nature. The subpar performance of RoBERTa compared to FastText despite it being a contextual model is one of the limitations of our work. When we are obtaining the course embeddings, we are not taking advantage of the contextual nature of the model to the fullest extent. Usually, a sentence is passed to the contextual model to obtain a word embedding using the contexts of the sentence. In our case, to get a course embedding, we could pass in an actual transaction sequence containing the course. We could also use Set Transformer as an additional model of comparison for course embedding, given our finding that the order of course sequences did not matter[7]. Next, future work could also focus on investigating why wait-listed transactions don't provide additional signals on top of enrollment, where as dropped, or dropped and waitlisted actions do add additional signal. Perhaps students are more likely to drop a course as they enroll in an equivalent course, than waitlisting a course as they enroll in another course that satisfies the same requirement. Lastly, while transaction data is shown to predict course features well, we could also use it to predict student-level features. For instance, we could explore the rationality of student decision-making by using additional transaction data features such as reason for enrollment actions.

8. CONCLUSION

Our study investigates the utility of novel transaction data (which contains granular non-enrollment student actions and chronologically ordered records) in similarity-based course recommendations. We evaluate such similarity signals with skip-gram, FastText, and RoBERTa models. We showed that transaction records including enrolling, waitlisting, and dropping student actions improve course similarity signals from enrollment records. Additionally, we found that chronology does not provide more course similarity signal than randomization of transaction records for skip-gram and FastText, but does so for RoBERTa. In fact, the best-performing model is FastText trained on random enrolling and dropping transactions. Our study provides some new pieces of information that could help course recommendation systems. We now know that chronology of enrollment is not beneficial to course2vec using skip-grams or FastText, but does benefit the transformer-based RoBERTa. We also found transaction course embeddings have greater predictive power in classifying courses into features such as STEM/non-STEM designation, department, and requirement satisfaction. The accuracy from predicting which courses satisfy major requirements significantly improves by using transactions (enroll and drop events) – from 81.6% to 95.0%, which is likely close to human advisor-level fidelity. This increase could be essential for course recommender models that may want to learn degree requirements from data.

9. REFERENCES

[1] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[2] M. H. de Menéndez, R. Morales-Menendez, C. A. Escobar, and R. A. R. Mendoza. Learning analytics: state of the art. *International Journal on Interactive Design and Manufacturing*, 16:1209–1230, 2022.

[3] H. Heuer and A. Breiter. Student success prediction and the trade-off between big data and data minimization. *DeLFI*, 2018.

[4] L. Huang, C.-D. Wang, H.-Y. Chao, J.-H. Lai, and P. S. Yu. A score prediction approach for optional course recommendation via cross-user-domain collaborative filtering. *IEEE Access*, 7:19550–19563, 2019.

[5] M. E. Ibrahim, Y. Yang, D. L. Ndzi, G. Yang, and M. Al-Maliki. Ontology-based personalized course recommendation framework. *IEEE Access*, 7:5180–5199, 2019.

[6] W. Jiang and Z. A. Pardos. Evaluating sources of course information and models of representation on a variety of institutional prediction tasks. In *Educational Data Mining*, 2020.

[7] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3744–3753. PMLR, 09–15 Jun 2019.

[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. cite arxiv:1907.11692.

[9] R. Morsomme and S. V. Alferez. Content-based course recommender system for liberal arts education. In *Educational Data Mining*, 2019.

[10] Z. A. Pardos, H. Chau, and H. Zhao. Data-assistive course-to-course articulation using machine translation. In *Proceedings of the Sixth (2019) ACM Conference on Learning @ Scale*, L@S '19, New York, NY, USA, 2019. Association for Computing Machinery.

[11] Z. A. Pardos, Z. Fan, and W. Jiang. Connectionist recommendation in the wild: On the utility and scrutability of neural networks for personalized course guidance. *User Modeling and User-Adapted Interaction*, 29(2):487–525, apr 2019.

[12] Z. A. Pardos and W. Jiang. Designing for serendipity in a university course recommendation system. In *Proceedings of the Tenth International Conference on Learning Analytics amp; Knowledge*, LAK '20, page 350–359, New York, NY, USA, 2020. Association for Computing Machinery.

[13] Z. A. Pardos and A. J. H. Nam. A university map of course knowledge. *PLOS ONE*, 15(9):1–24, 09 2020.

[14] S. Reddy, I. Labutov, and T. Joachims. Latent skill embedding for personalized lesson sequence recommendation. *CoRR*, abs/1602.07029, 2016.

[15] N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[16] M. C. Urdaneta-Ponte, A. Mendez-Zorrilla, and I. Oleagordia-Ruiz. Recommendation systems for education: Systematic review. *Electronics*, 10(14), 2021.

[17] N. N. Vo, Q. T. Vu, N. H. Vu, T. A. Vu, B. D. Mach, and G. Xu. Domain-specific nlp system to support learning path and curriculum design at tech universities. *Computers and Education: Artificial Intelligence*, 3:100042, 2022.

[18] Y. Xu and Z. A. Pardos. Extracting course similarity signal from enrollments using subword embeddings. Under review at the 18th Workshop on Innovative Use of NLP for Building Educational Applications, Submitted.

[19] G. Zhu, N. A. Kopalle, Y. Wang, X. Liu, K. Jona, and K. Börner. Community-based data integration of course and job data in support of personalized career-education recommendations. *Proceedings of the Association for Information Science and Technology*, 57, 2020.

APPENDIX

A. OPTIMAL MODEL HYPERPARAMETERS

A.1 FastText

The optimal hyperparameters for the best performing model (FastText trained on randomly ordered E & D transaction records) are as follows: min count = 50, window = 9, vector size = 210, sample = 3.e-05, alpha = 0.04, min alpha = 0.0007, negative = 15.

A.2 Predictive Models

The hyperameters used for the multinomial logistic regression are: max number of iterations = 1000, penalty = l2 norm. The hyperparameters used for MLP are as follows: hidden layer = 100, activation function = relu, solver = adam, alpha = 0.0001, batch size = min (200, number of samples), learning rate = 0.001, maximum number of iterations = 200.

Predicting Bug Fix Time in Students' Programming with Deep Language Models

Stav Tsabari
Ben-Gurion University
stavts@bgu.ac.il

Avi Segal
Ben-Gurion University
avise@post.bgu.ac.il

Kobi Gal
Ben-Gurion University
University of Edinburgh
kobig@bgu.ac.il

ABSTRACT

Automatically identifying struggling students learning to program can assist teachers in providing timely and focused help. This work presents a new deep-learning language model for predicting "bug-fix-time", the expected duration between when a software bug occurs and the time it will be fixed by the student. Such information can guide teachers' attention to students most in need. The input to the model includes snapshots of the student's evolving software code and additional meta-features. The model combines a transformer-based neural architecture for embedding students' code in programming language space with a time-aware LSTM for representing the evolving code snapshots. We evaluate our approach with data obtained from two Java development environments created for beginner programmers. We focused on common programming errors which differ in their difficulty and whether they can be uniquely identified during compilation. Our deep language model was able to outperform several baseline models that use an alternative embedding method or do not consider how the programmer's code changes over time. Our results demonstrate the added value of utilizing multiple code snapshots to predict bug-fix-time using deep language models for programming.

Keywords

computer programming, predict bug fix time, deep learning language models

1. INTRODUCTION

Programming courses have become an essential component of many STEM degrees and are attracting students from diverse backgrounds. Many beginners struggle with learning fundamental principles of programming [20]. Additionally, previous studies suggest that compiler messages have an imperfect mapping to errors which can confuse the students [24]. Providing students with personalized support can significantly aid their learning. The time spent by programmers to fix bugs is known to be a proxy for the difficulty they encounter and can be used as an indicator for finding struggling students [14, 3]. Thus, predicting the bug-fix-time of errors for students can help teachers identify those students requiring additional attention and support. Such prediction can also support hint generation systems for better inferring when a hint is needed [11, 23].

Past work has estimated the bug-fix-time for different errors based on bug error reports. These are reports created by the quality assurance team in organizations to describe and document the bugs found in computer programs [15, 33, 18]. These studies ignored the personal variations between programmers, predicting a single value per a specific bug.

We address this gap by providing a personalized approach for predicting bug-fix-time for programming errors. Our underlying assumption is that if the student's fix time for a bug is longer than a threshold, it may indicate a struggling student requiring assistance and guidance. Specifically, our method predicts if the error fix time will be "short" or "long", with the median used as the cutoff value. The median is chosen as threshold to focus on the lower half of students that may benefit from some level of assistance. This is a standard approach in other works studying bug-fix-time [5, 15].

Our approach is personalized per student and per bug type and uses snapshots of the evolving student's code. Errors vary in whether the compiler can identify the error, and whether the compiler's error message is unique to the specific error type. The proposed method is based on CodeBert[13], a state-of-the-art transformer-based neural architecture for embedding students' programming code, and combines an LSTM-based architecture which is used to capture multiple time dependant code snapshots.

We compared our approach to three baselines for predicting the fixed time of the different errors in two datasets (1) A method that is based on the Halstead Metrics [16]. This approach computes features based on operators and operands in the code. (2) A code embedding-based approach using Code2Vec [2], which is a common framework for learning representations of natural language and code, and has been used previously in an educational context. This approach considered the student's code which produced the bug as well as the prior code submissions of the same program. (3) A language model-based approach using CodeBert which considered only the student's code that produced the bug (4) Our approach: A language model-based approach using

S. Tsabari, A. Segal, and K. Gal. Predicting bug fix time in students' programming with deep language models. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 396–405, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115733

CodeBert which considered the student's code which produced the bug as well as the previous code snapshots saved by the system while the student was evolving their code.

We evaluated our approach on code and compilation instances obtained from thousands of students' code submissions, sampled from two different programming environments. The first environment was the BlueJ Java development environment [21], a programming environment designed for beginner programmers and used in a large number of educational institutions. We obtained 241,418 code submission instances containing errors that were generated by students when learning to program. The prediction task focused on 4 common types of novice errors that differ in complexity.

The second environment, called CodeWorkout, was collected from an introductory programming course in the Spring and Fall of 2019 semesters at a public university in the U.S. [12]. We obtained 80,013 code submission instances that contained 75 different compilation errors, each error has a different cause, such as: unknown variable, missing operands etc. The CodeWorkout dataset contained simpler computing problems, with typically shorter submitted programming solutions.

We considered two different settings for the BlueJ dataset, one in which a different model was built for each error type and a setting where one model was built for all error types. The CodeWorkeout dataset was tested with one model per all error types due to data size limitations.

For both environments, our proposed approach was able to outperform the baseline approaches in terms of accuracy, recall, and F1 measures when predicting bug-fix-time. These results demonstrate the efficiency of using transformer-based language models developed for programming languages for solving the bug-fix-time prediction task and the value of adding students' code history for such tasks. Our approach has implications for software development education, in that it can potentially be used by instructors to identify struggling students requiring further support.

2. RELATED WORK

Our work relates to several research areas: (1) Programming errors performed by novice programmers (2) Predicting bug-fix-time for programming errors, and(3) Recent deep Natural Language Processing language models for programming language representations. We elaborate on each one in turn.

2.1 Student Programming Errors

Hristova et al. [19] collected a list of common students' Java programming errors based on reporting of teaching assistants. Most of the errors identified were detected and reported by the Java compiler. Nonetheless, McCall et al. [24] investigated logical errors in students' code and suggested that compiler messages alone have an imperfect mapping to student logical errors. They demonstrated that the same logical error can produce different compiler error messages and different logical errors can produce the same compiler error message. Bayman et al. [8] examined errors related to individual program statements and found that many learners possess a wide range of misconceptions about individual statements or constructs of even very simple statements, which lead to programming errors.

Brown et al. [4] analyzed 18 students' errors using the BlueJ dataset and focused on two error types, those identified by the compiler, and those that require a customized source code analyzer that searches the source code for programming mistakes. Errors relating to the latter type are not uniquely identifiable by the compiler and may be more complex to fix. We are inspired by this study and test the potential of machine learning-based approaches to predict the bug-fix-time of student errors for both compiler errors as well as errors identified by static source code analysis.

2.2 Predicting Bug Fix Time of Programming Errors

Various approaches have been used in past research to predict the time required for fixing bugs. Zhang et al. [32] investigated the connection between bug reports and other features and the bug fixing time. Bug reports are the reports created by the quality assurance and testing team in an organization to describe and document the bugs found in a computer program and include attributes such as problem description and priority. Zhang et al. [33] predicted the number of bugs to be fixed and estimated the time required to fix a certain bug using bug report attributes only. They estimated the time to fix a bug as "slow" or "quick" based on several thresholds. Other studies have focused on predicting bug fixing time using different classifications than "slow" or "fast". Panjer et al. [25] employed multi-classification using various classification models to classify the time to fix bugs into seven-time buckets, using only the bug report.

Some studies have focused on predicting the exact time to fix the bug. Weiss et al. [29] used text similarity to predict the bug-fixing time. Given a new bug report, they used text similarity to search for similar, earlier reports and use their average time as the prediction time. Recently, some deep network-based approaches were proposed for the bug-fix-time prediction problem. Ardimento et al. [5] used BERT, a pre-trained deep bidirectional Transformer model, to predict bug fixing time as fast or slow from bug reports. This approach has shown the best performance so far.

Our research differentiates from these past efforts in three main manners: (1) First, all past work performed non personalized bug-fix-time prediction. I.e., the prediction was performed per error type and not per user. In contrast, we focus on predicting bug-fix-time per user for each error type. (2) Second, past studies used errors of experienced programmers and were trained on code repositories such as GitHub and the like. In this research, we focus on errors generated by novice student programmers and use appropriate datasets for this task. (3) Third, past studies did not directly take into account the programmer's source code nor did they use previous code snapshots which capture the programmer's evolving code prior to the error generation. Specifically, these works used only attributes from bug reports and did not directly consider the code in which the bug was found. In this paper, we hypothesize that the source code itself as well as past code snapshots of the programmer's evolving work hold strong signals for predicting the bug-fix-time for errors generated by the programmer.

2.3 Language Models for Programming Language Representations

One technique for the embedding of software program methods is Code2Vec, a neural model for representing snippets of code as continuously distributed vectors [2]. Code2Vec was developed for the task of method name prediction and uses paths in the program's abstract syntax tree (AST) for its embeddings [7]. We choose to use Code2Vec as a baseline since it outperformed other models in past works and has been used previously in educational contexts [26, 6].

Recently, deep language models have been developed for code representations. One such model which demonstrated state-of-the-art results is CodeBert [13]. CodeBert is a transformer based large scale language model for both natural and programming languages. The model is trained with a dataset that includes 6.4M unimodal source codes in different programming languages including Java, Python, Go, JavaScript, PHP, and Ruby. CodeBert learns general-purpose representations and can be fine-tuned to support downstream natural language and programming language applications such as source code classification tasks. We used CodeBert for code representation in our proposed approach.

3. METHODOLOGY

In this study, we evaluated the usage of a large scale deep learning language model combined with software code snapshots for the prediction of bug fix time. Specifically, our research questions were as follows: (**RQ1**) Do models that embed students' code do better than those relying on Halstead metrics features when predicting bug fix time? (**RQ2**) Can deep language models built for software code representation improve such a prediction task? (**RQ3**) Does using preceding snapshots of the students' code further improves the prediction task?

3.1 Datasets

To increase the generality of the developed approach, two datasets were used in this study, both from development environments created for the Java programming language. The first dataset was obtained from the BlueJ environment which is a general-purpose Java programming environment designed for beginner programmers. The second dataset was collected on the CodeWorkout environment which contains assignments submitted by Java students during two semesters. We note that the obtained data from both programming environments did not include any personal or demographic information about users. In both datasets, we leave out errors that were not solved altogether by the student (i.e., the bug fix time is unknown).

3.1.1 The BlueJ Dataset

BlueJ is an integrated Java programming environment designed for beginners and used in a large number of institutions around the world [21]. The environment has been used for a variety of assignments designed to support and exploit pedagogical theories of programming. The BlueJ platform includes an advanced capability of recording the student's programs (as they are being developed) in a dedicated research environment called Blackbox [9]. In Blackbox, each instance includes a timestamp of a compilation event by a programmer, together with the code that was submitted for that compilation, a student ID, a session ID, and a list of error messages reported by the compiler (if any).

Table 1: Selected Errors in the BlueJ Dataset

Bug Type	***Median BFT(sec.)***	***Average BFT(sec.)***	***STD BFT(sec.)***	***Num. instances***
I	51	164	261	80,000
O	35	136	239	80,000
A	52	193	298	60,254
B	60	228	398	21,164

Similarly to Brown et al.[4], we used a dataset representing one year of activity in the BlueJ environment, from September 1st, 2013 to August 31st, 2014. In this set, we focus on the four errors that were identified by Brown et al.[4] among the most common errors for novice programmers:

Error I: Invoking method with a wrong argument type; the compiler can uniquely detect this error.
Error O: Non-void method without a return statement; the compiler can uniquely detect this error.
Error A: Confusing operator (=) with (==); the compiler detects error, but does not output a unique error statement.
Error B: Using the operator (==) instead of (.equals); the compiler cannot detect the error.

In this research, errors I and O were identified directly from the output messages issued by the compiler. Errors A and B were identified by a static analyzer built using XML representation [10] of projects' code. In total, the dataset contains $17,682,006$ instances. From this dataset, we sample $241,418$ instances which include the four mentioned error types. Each instance in the dataset is a code submission.

Table 1 presents the number of instances and the bug-fix-time (BFT) median, average, and STD values for each selected error in the BlueJ platform. As shown by the table, the errors vary in average difficulty in terms of the average fix time. An example of the distribution of bug-fix-time for error I in BlueJ can be seen in Figure 1. As seen in the figure, the distribution is right-skewed, with some students exhibiting very long fix times for this error. A similar trend was also apparent for the other bug types in the platform.

3.1.2 The CodeWorkout Environment

The CodeWorkout environment [12] is an online system for people learning Java programming for the first time. This open-source site contains 837 coding problems spanning topics such as sorting, searching, and counting. The environment includes tests for each problem, which verifies the correctness of each student's submission. Student submissions are graded automatically using these tests and feedback is returned including error messages.

The dataset includes student assignment submissions from an introductory course of Computer Science course ("CS1") administered in the Spring and Fall 2019 semesters at a public university in the U.S. During this course, 50 different coding problems from CodeWorkout were given to students,

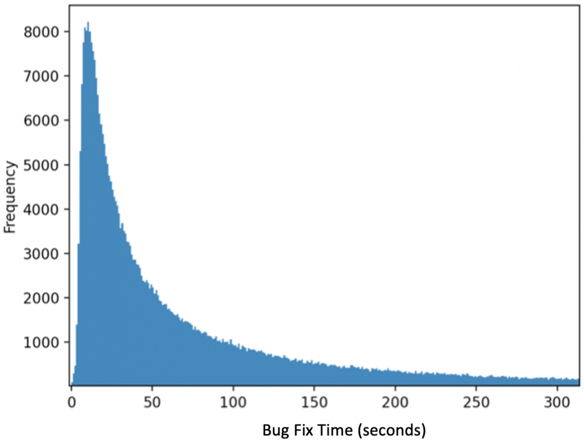


Figure 1: Distribution of bug-fix-time for Error I in BlueJ

and their code submissions were collected. The coding problems used in this dataset are easy and designed for beginner programmers. Each problem requires 10-26 lines of code and includes basic operations such as for loops, and if / else statements. Each submission in the CodeWorkout dataset includes the submitted code by the user, the user ID, the assignment ID, and the list of reported compiler errors if any. The CodeWorkout dataset included only bugs that are identifiable by the compiler. Overall this dataset contains 75 error types from 819 students and a total of 80,013 instances. As with the BlueJ dataset, the bug-fix-time in the CodeWorkout datasets exhibits a long tail distribution.

3.2 Computing Bug-Fix-Time

The bug-fix-time (BFT) for a given error is defined as the length of time (in seconds) between the first compilation submission in which the error was reported, and the nearest compilation submission in which the error was resolved. For the BlueJ environment, The bug-fix time is adjusted to the periods when the user is logged into the system. For CodeWorkout, session login information is not available, so bug-fix-time considers only bug occurrence and bug resolution timestamps.

We note the high variance in bug fixing time as indicated by the STD value in table 1 (also occurs for CodeWorkout). This indicates that bug-fixing time is a personal phenomenon, which may depend on specific students' characteristics. We hypothesize that such characteristics are expressed in the way that students write and evolve their program code. Thus, we define the bug-fix-time prediction problem, as the problem of predicting the time to fix a bug for a **specific student** given the occurrence of a **specific bug** in their latest compilation and considering their past code submissions. Following past approaches, we use the median value of bug fixing time per each bug type as the threshold between "slow" and "fast" fixing times [5].

3.3 Predicting Bug Fix Time

We now describe the deep learning model for the prediction task at hand. Our architecture includes three layers: (1) An embedding layer using pre-trained deep language models for representing programming languages, (2) A time-aware layer using Long Short Term Memory, and (3) A classification layer. The architecture can be seen in Figure 2.

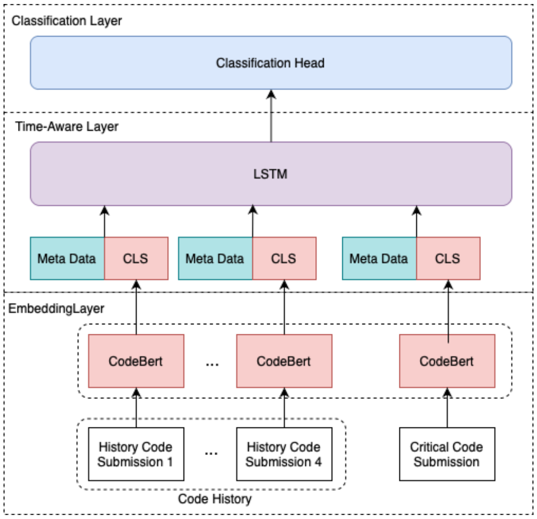


Figure 2: Model Architecture

The input to the model is a student's compilation submission that includes the following:

First, the Critical Code snapshot: the code snapshot that generated the error. Second, the Code History: the most recent code submissions that preceded the critical code submission. We use 4 preceding code submissions as this is the median number of available code snapshots before an error is identified, in both datasets[1]. If the code history was shorter, we used zero-based representations for the empty submissions.

Third, we added four meta features relating to the user. These include: (a) The number of compilation submissions performed before the error occurred. Represents how long the user is working on this program. A higher number of submissions may indicate a struggling or hesitant student. (b) A binary value indicating whether the student generated this error before in any of their previous submission in blueJ. If the student had seen this error before, it may be easier for them to solve this error. (c) A binary value indicating whether the compiler has detected additional errors at the same compilation. Multiple errors may indicate that the student is struggling and will need a longer time to fix the designated error. (d) A value indicating the user experience in the system. For BlueJ, this is the time since the user created the account (available only on BlueJ). For CodeWorkout this is the number of assignments the user has submitted out of the total assignments given in the university course used for the dataset.

We note that if a code submission generated multiple errors, we created multiple instances, one for each error type generated by this code submission. Additionally, we have tried meta-features b,c and d as integers and as binary values and used the representation that had the best results.

[1] We explore the sensitivity of the results to the code history length in Appendix A.

Embedding layer. The first model's layer encodes student's code submission and the history of previous code submissions. We use CodeBert [13] for this embedding layer. CodeBert is a bimodal pre-trained language model for programming languages and natural language text comments. It is based on the RoBERTa-base [22] model architecture, a BERT-based language model with 12 transformer layers, 768-dimensional hidden states, and 12 attention heads. The input format to CodeBert is concatenating two data segments with a special separator token, namely [CLS]. The first segment is natural language text representing the comments in the program and the second segment is the programming code itself. The output of CodeBert includes a representation of the [CLS] token which works as the aggregated representation for the input code snapshot. This output is a 768-dimensional vector and is passed from the embedding layer to the next layer. We note that the maximum input sequence length of CodeBert is 512 tokens. Longer snapshots are truncated to the first 512 tokens[2].

Time-Aware layer. The time-aware layer is designed to reflect the changes in the user code over time. This layer utilizes a Bidirectional Long Short Term Memory (LSTM) [17]. Each code snapshot representation from the previous layer is concatenated to additional four features about the user added to the end of the code snapshot representation. This results in a 772-dimensional vector fed into the Bidirectional LSTM layer. The output of the Bidirectional LSTM layer is a 1544-dimensional vector.

Classification layer. The last layer is the classification layer designed to predict the binary bug-fix-time value (slow or fast). This layer takes the output of the LSTM layer and feeds it into the following layers: (a) A fully connected layer with an output size of 128. This fully connected layer multiplies the input by a weight matrix and then adds a bias vector, using the relu activation function (2) A fully connected layer with an output size of 2 and (3) A Sigmoid function. The output is a binary prediction score of slow or fast time-to-fix.

3.4 Baselines

We evaluated our model against alternative approaches that vary in how students' code is represented and whether the history of past code compilations is considered.

Halstead Metrics Based Method. This baseline is the Halstead metrics method used for measuring code. The method views a computer program as a collection of operator and operand tokens and proposes 12 metrics as described in [16]. In this baseline, we represent each code snapshot with a 12-dimensional vector based on Halstead metrics. This embedding replaces the CodeBert embedding in figure 2. The LSTM in this method is fed with a 16-dimensional vector for each snapshot.

Code2Vec Based Method. This baseline used Code2Vec [2], an Abstract Syntax Tree (AST) [27] embedding model for code. We used the pre-trained model of Code2Vecv from [2]. The Code2Vec embedding replaces the CodeBert embedding in figure 2.

Critical Code only. This baseline used a version of our proposed model which does not consider past snapshots of the programmer's evolving code. For this baseline, only the critical code submission and the 4 additional meta-features for this submission are used and the LSTM layer is removed.

4. EXPERIMENTS

To address the research questions, four different methods were compared during the experiments:

Halstead Metric Based Method: predicts bug fix time using the code snapshot that contains the error and four preceding snapshots (i.e. code history). Each code snapshot is represented using the Halstead metrics and 4 additional user features (used for RQ1).
Code2Vec Based Method: predicts bug fix time using the snapshot that contains the error and four preceding snapshots. Each code snapshot was embedded using Code2Vec and 4 additional user features (used for RQ2).
Critical code only: predicts bug fix time using the full code snapshot that contains the error and additional 4 features (used for RQ3).
Proposed model: predicts bug fix time using the snapshot that contains the error and four preceding snapshots, each one embedded using CodeBert and 4 additional meta-features.

Our experiments evaluate the above approaches in two different setups: (1) Error-specific: in which a prediction model is trained and evaluated for each error type in separation, and (2) Error-agnostic: where one prediction model is trained and evaluated for multiple error types.

Unfortunately, the CodeWorkout dataset contains on average only 455 instances per error type, so there is not enough data for the error-specific approach for this dataset. Thus, only the error-agnostic model was evaluated for this dataset. All experiments were evaluated using a 5-Fold cross-validation setup and the recommended hyperparameters values from the literature.

The metrics used include: (1) ROC-AUC: summarizes how well the model separates the positive and negative samples for different thresholds. (2) Recall (positive samples): the ratio of positive samples correctly classified as positive to the total number of positive samples. (3) F1 (positive samples): combines the precision (i.e. number of true positive results divided by the number of all positive results) and recall of a classifier into a single metric by taking their harmonic mean.

We focus on the positive samples which represent struggling students that took a long time to fix a bug. Therefore, recall and F1 metrics are measured and reported for this class.

Statistical significance was tested for all results using the Wilcoxon signed rank test [30]. Post-hoc corrections for statistical tests were performed using the Holm-Bonferroni

[2] We explore the sensitivity to other truncation approaches in Appendix A.

Table 2: BlueJ - Error Specific Results

Results for Error I			
Method	***Recall[%]***	***F1*[%]**	***ROC-AUC[%]***
Halstead Metric Based	44	49	55
Code2Vec Based	57	56	56
Critical Code Only	64	59	55
Proposed model	**74***	**64***	**62***

Results for Error O			
Method	***Recall[%]***	***F1*[%]**	***ROC-AUC[%]***
Halstead Metric Based	49	52	56
Code2Vec Based	59	56	54
Critical Code Only	57	55	53
Proposed model	**75***	**63***	**60***

Results for Error B			
Method	***Recall[%]***	***F1*[%]**	***ROC-AUC[%]***
Halstead Metric Based	44	46	49
Code2Vec Based	53	50	49
Critical Code Only	63	56	51
Proposed model	**83***	**64***	**54***

Results for Error A			
Method	***Recall[%]***	***F1*[%]**	***ROC-AUC[%]***
Halstead Metric Based	42	46	50
Code2Vec Based	55	52	51
Critical Code Only	60	57	55
Proposed model	**70***	**64***	**61***

method [1]. A star mark ("*") in the results tables (tables 2, 3, 4) denotes a model is significantly better than the rest.

4.1 Error-Specific Results

Table 2 displays the results for error-specific models in BlueJ. As seen in the table, the proposed model outperforms all other approaches for all error types and for all measured metrics. Interestingly, the Code2Vec baseline did not improve over the Halstead metric method in some of the error types on ROC-AUC metric.

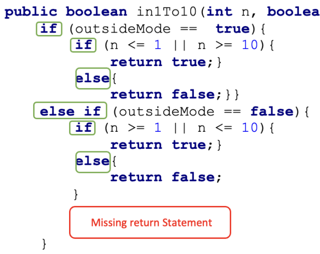
4.2 Error-Agnostic Results

We compare models' performance in the error-agnostic case:

BlueJ Dataset. On the BlueJ dataset, we combined the four errors A, B, I and O. The dataset contained 80,000 instances (sampled from the entire dataset). For each instance in the dataset, the binary labels were determined separately for each error type. Table 3 presents the performance of this dataset. As seen in the table, the proposed method outperformed all baselines in all measured metrics. The second performing approach was the Critical Code Only approach. Code2Vec embedding showed better results than the method based on the shallow Halstead metrics embedding [3].

Table 3: BlueJ - Error Agnostic Results

Results for Errors A+B+I+O			
Method	***Recall[%]***	***F1*[%]**	***ROC-AUC[%]***
Halstead Metric Based	47	48	48
Code2Vec Based	55	54	52
Critical Code Only	61	56	55
Proposed model	**77***	**64***	**62***



```
public boolean in1To10(int n, boolean outsideMode){
    if (outsideMode ==  true){
        if (n <= 1 || n >= 10){
            return true;}
        else{
            return false;}}
    else if (outsideMode == false){
        if (n >= 1 || n <= 10){
            return true;}
        else{
            return false;
        }
        Missing return Statement
    }
```

Figure 3: Correct Prediction Using Code Text

CodeWorkout Dataset. For the CodeWorkout dataset, we combined code submissions for all 75 error types into one dataset that contained 80,013 instances. The binary labels were determined based on the median bug-fix-time threshold for each error type in separation. Table 4 presents the results for this dataset. As seen in the table, the proposed model outperformed all other baselines on all measured metrics. The second performing model was the Critical Code Only model. For this dataset, the Code2Vec-based model outperformed the Halstead-based model in 2 of the 3 metrics.

Table 4: CodeWorkout - Error Agnostic Results

Method	***Recall[%]***	***F1*[%]**	***ROC-AUC[%]***
Halstead Metric Based	54	55	52
Code2Vec Based	58	55	56
Critical Code Only	65	62	65
Proposed model	**70***	**64***	**70***

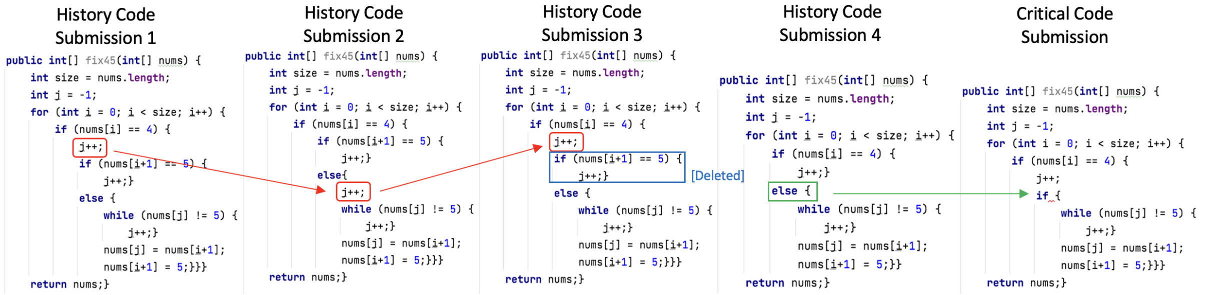
5. CASE STUDIES

To further demonstrate the performance of the proposed method, we present two illustrative examples.

5.1 Case A: The Value of Code Text

Figure 3 presents a code submission that contains the error "Missing Return Statement". The user generating this error took a long time to fix. While the model that used the Halstead metrics was wrong in predicting a "short" label for this snapshot, the two CodeBert-based models performed a correct prediction. As seen in the figure, the submitted code contains multiple if-else statements which may make it difficult for the student to identify that yet another return statement is missing and its location. We hypothesize that a code-based model correctly classified this sample since it is

[3] We evaluate feature importance for the proposed model in Appendix B.



History Code Submission 1

```
public int[] fix45(int[] nums) {
    int size = nums.length;
    int j = -1;
    for (int i = 0; i < size; i++) {
        if (nums[i] == 4) {
            j++;
            if (nums[i+1] == 5) {
                j++;}
            else {
                while (nums[j] != 5) {
                    j++;}
                nums[j] = nums[i+1];
                nums[i+1] = 5;}}}
    return nums;}
```

History Code Submission 2

```
public int[] fix45(int[] nums) {
    int size = nums.length;
    int j = -1;
    for (int i = 0; i < size; i++) {
        if (nums[i] == 4) {
            if (nums[i+1] == 5) {
                j++;}
            else{
                j++;
                while (nums[j] != 5) {
                    j++;}
                nums[j] = nums[i+1];
                nums[i+1] = 5;}}}
    return nums;}
```

History Code Submission 3

```
public int[] fix45(int[] nums) {
    int size = nums.length;
    int j = -1;
    for (int i = 0; i < size; i++) {
        if (nums[i] == 4) {
            j++;
            if (nums[i+1] == 5) {
                j++;}
            else {
                while (nums[j] != 5) {
                    j++;}
                nums[j] = nums[i+1];
                nums[i+1] = 5;}}}
    return nums;}
```

History Code Submission 4

```
public int[] fix45(int[] nums) {
    int size = nums.length;
    int j = -1;
    for (int i = 0; i < size; i++) {
        if (nums[i] == 4) {
            j++;}
        else {
            while (nums[j] != 5) {
                j++;}
            nums[j] = nums[i+1];
            nums[i+1] = 5;}}
    return nums;}
```

Critical Code Submission

```
public int[] fix45(int[] nums) {
    int size = nums.length;
    int j = -1;
    for (int i = 0; i < size; i++) {
        if (nums[i] == 4) {
            j++;
            if {
                while (nums[j] != 5) {
                    j++;}
                nums[j] = nums[i+1];
                nums[i+1] = 5;}}}
    return nums;}
```

Figure 4: Correct Prediction Using the Code History

using the entire code structure, while a shallow model relying only on item counting is blind to such subtle differences.

5.2 Case B: The Value of Code History

Figure 4 presents code submissions that contain an error that brackets are missing with a binary label of "long". While the model that used only the last snapshot (Critical Code Only model) predicted a "short" fix time, the proposed model predicted correctly that the fix time will be "long". Looking at the code submission history may explain why the student is struggling and why it took them a long time to solve the error. As shown in code submissions 1-3 in red, the student changed the code and then changed it back. In code submission 4 they then deleted a full "if" statement and changed an "else" statement to an "if" statement which led to the error. This behavior is most likely a behavior of a struggling student and may indicate that when faced with a resulting error, it will take them a long time to fix it. Such inference can only be captured by a model which considers multiple snapshots and tracks the student's behavior over time.

6. DISCUSSION

The results of this study demonstrate that deep language models built for code representation can significantly improve on past models when predicting bug-fix time (RQ2). They also show that using multiple code snapshots further improves such results (RQ3) validating the benefits of combining the latest language models built for code representation with a multi-snapshot approach. These results reflect the representation power of the latest language models, which are pre-trained on vast amounts of past data. Interestingly, the simpler Halstead method-based approach outperformed the Code2Vec approach in some cases (RQ1). This demonstrates that earlier deep learning methods (such as Code2Vec), which were trained on fewer data and with less sophisticated neural network structures, lack the power inherited in the latest approaches. The key takeaway of these findings is the potential and importance of harnessing such latest language models in the educational data mining field, as it relates to software education and beyond.

We mention some limitations of the proposed approach. First, the computed bug fix time is only an estimation of the true, latent value of the actual time spent by a user on fixing a bug. Even when sign-in and sign-out information is available, such as in the BlueJ dataset, the user may have been occupied with other activities when logged in, contrary to our assumptions. Second, the model assumes each error is independent even though one error can lead to another error. Third, the CodeBert model, similar to other Bert-based models, is limited to 512 input tokens. We used truncation approaches to accommodate this limitation. Nonetheless, future work may consider other approaches (such as summarization, hierarchical representation, etc.) to accommodate longer code snapshots. Fourth, the rapid improvement in language models for code representation implies that CodeBert is only an early bird among an increasing number of evolving models in the field [31]. As such, additional latest models should be investigated in future work.

7. CONCLUSION AND FUTURE WORK

This work provides a new approach for predicting whether a student's bug-fix-time will be "short" or "long" based on a given threshold for common errors made by novice programmers. Predicting a "long" bug-fix-time is one possible way to identify struggling students in need of teacher support. We developed and compared four approaches towards this task (1) A model using Halstead metrics computed over multiple code snapshots preceding a software error (2) A model using Code2Vec for code embedding that considers the code compilation which produced the error and previous student's code snapshots (3) A model using CodeBert for code embedding which considers only the code compilation which produced the error (4) Our approach: a model using CodeBert for code embedding which considers the code submission producing the error and previous student's code submissions. Our approach was able to outperform all baselines for ROC-AUC, Recall, and F1. Our results demonstrate the efficacy of CodeBert and of using multiple time-based code snapshots in identifying struggling students by predicting the bug-fix-time of their software errors.

In future work, we intend to cover additional common student errors and extend this study to different programming languages. Furthermore, during data pre-processing, we found out that some errors are not solved by some students altogether and we plan to extend our model to identify such cases. Finally, we are working on developing and evaluating a regression-based model to predict a continuous bug-fix-time value to better estimate how long it will take students to solve their programming errors.

8. ACKNOWLEDGEMENTS

This work was supported in part by the European Union Horizon 2020 WeNet research and innovation program under grant agreement No 823783.

9. REFERENCES

[1] H. Abdi. Holm's sequential bonferroni procedure. *Encyclopedia of research design*, 1(8):1–8, 2010.

[2] U. Alon, M. Zilberstein, O. Levy, and E. Yahav. code2vec: Learning distributed representations of code. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–29, 2019.

[3] B. S. Alqadi and J. I. Maletic. An empirical study of debugging patterns among novices programmers. In *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education*, pages 15–20, 2017.

[4] A. Altadmri and N. C. Brown. 37 million compilations: Investigating novice programming mistakes in large-scale student data. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, pages 522–527, 2015.

[5] P. Ardimento and C. Mele. Using bert to predict bug-fixing time. In *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, pages 1–7. IEEE, 2020.

[6] D. Azcona, P. Arora, I.-H. Hsiao, and A. Smeaton. user2code2vec: Embeddings for profiling students based on distributional representations of source code. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 86–95, 2019.

[7] I. D. Baxter, A. Yahin, L. Moura, M. Sant'Anna, and L. Bier. Clone detection using abstract syntax trees. In *Proceedings. International Conference on Software Maintenance (Cat. No. 98CB36272)*, pages 368–377. IEEE, 1998.

[8] P. Bayman and R. E. Mayer. A diagnosis of beginning programmers' misconceptions of basic programming statements. *Communications of the ACM*, 26(9):677–679, 1983.

[9] N. C. C. Brown, M. Kölling, D. McCall, and I. Utting. Blackbox: A large scale repository of novice programmers' activity. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 223–228, 2014.

[10] M. L. Collard, M. J. Decker, and J. I. Maletic. srcml: An infrastructure for the exploration, analysis, and manipulation of source code: A tool demonstration. In *2013 IEEE International Conference on Software Maintenance*, pages 516–519. IEEE, 2013.

[11] Y. Dong, S. Marwan, P. Shabrina, T. Price, and T. Barnes. Using student trace logs to determine meaningful progress and struggle during programming problem solving. *International Educational Data Mining Society*, 2021.

[12] S. H. Edwards and K. P. Murali. Codeworkout: short programming exercises with built-in data collection. In *Proceedings of the 2017 ACM conference on innovation and technology in computer science education*, pages 188–193, 2017.

[13] Z. Feng, D. Guo, D. Tang, N. Duan, X. Feng, M. Gong, L. Shou, B. Qin, T. Liu, D. Jiang, et al. Codebert: A pre-trained model for programming and natural languages. *arXiv preprint arXiv:2002.08155*, 2020.

[14] S. Fitzgerald, G. Lewandowski, R. McCauley, L. Murphy, B. Simon, L. Thomas, and C. Zander. Debugging: finding, fixing and flailing, a multi-institutional study of novice debuggers. *Computer Science Education*, 18(2):93–116, 2008.

[15] E. Giger, M. Pinzger, and H. Gall. Predicting the fix time of bugs. In *Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering*, pages 52–56, 2010.

[16] M. H. Halstead. Natural laws controlling algorithm structure? *ACM Sigplan Notices*, 7(2):19–26, 1972.

[17] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[18] P. Hooimeijer and W. Weimer. Modeling bug report quality. In *Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering*, pages 34–43, 2007.

[19] M. Hristova, A. Misra, M. Rutter, and R. Mercuri. Identifying and correcting java programming errors for introductory computer science students. *ACM SIGCSE Bulletin*, 35(1):153–156, 2003.

[20] T. Jenkins. On the difficulty of learning to program. In *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, volume 4, pages 53–58. Citeseer, 2002.

[21] M. Kölling, B. Quig, A. Patterson, and J. Rosenberg. The bluej system and its pedagogy. *Computer Science Education*, 13(4):249–268, 2003.

[22] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[23] Y. Mao, Y. Shi, S. Marwan, T. W. Price, T. Barnes, and M. Chi. Knowing" when" and" where": Temporal-astnn for student learning progression in novice programming tasks. *International Educational Data Mining Society*, 2021.

[24] D. McCall and M. Kölling. Meaningful categorisation of novice programmer errors. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, pages 1–8. IEEE, 2014.

[25] L. D. Panjer. Predicting eclipse bug lifetimes. In *Fourth International Workshop on Mining Software Repositories (MSR'07: ICSE Workshops 2007)*, pages 29–29. IEEE, 2007.

[26] Y. Shi, Y. Mao, T. Barnes, M. Chi, and T. W. Price. More with less: Exploring how to use deep learning effectively through semi-supervised learning for automatic bug detection in student code. *International Educational Data Mining Society*, 2021.

[27] K. Slonneger and B. L. Kurtz. *Formal syntax and semantics of programming languages*, volume 340. Addison-Wesley Reading, 1995.

[28] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[29] C. Weiss, R. Premraj, T. Zimmermann, and A. Zeller. How long will it take to fix this bug? In *Fourth International Workshop on Mining Software Repositories (MSR'07: ICSE Workshops 2007)*, pages 1–1. IEEE, 2007.

[30] R. F. Woolson. Wilcoxon signed-rank test. *Wiley*

encyclopedia of clinical trials, pages 1–3, 2007.

[31] F. F. Xu, U. Alon, G. Neubig, and V. J. Hellendoorn. A systematic evaluation of large language models of code. In *Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming*, pages 1–10, 2022.

[32] F. Zhang, F. Khomh, Y. Zou, and A. E. Hassan. An empirical study on factors impacting bug fixing time. In *2012 19th Working conference on reverse engineering*, pages 225–234. IEEE, 2012.

[33] H. Zhang, L. Gong, and S. Versteeg. Predicting bug-fixing time: an empirical study of commercial software projects. In *2013 35th International Conference on Software Engineering (ICSE)*, pages 1042–1051. IEEE, 2013.

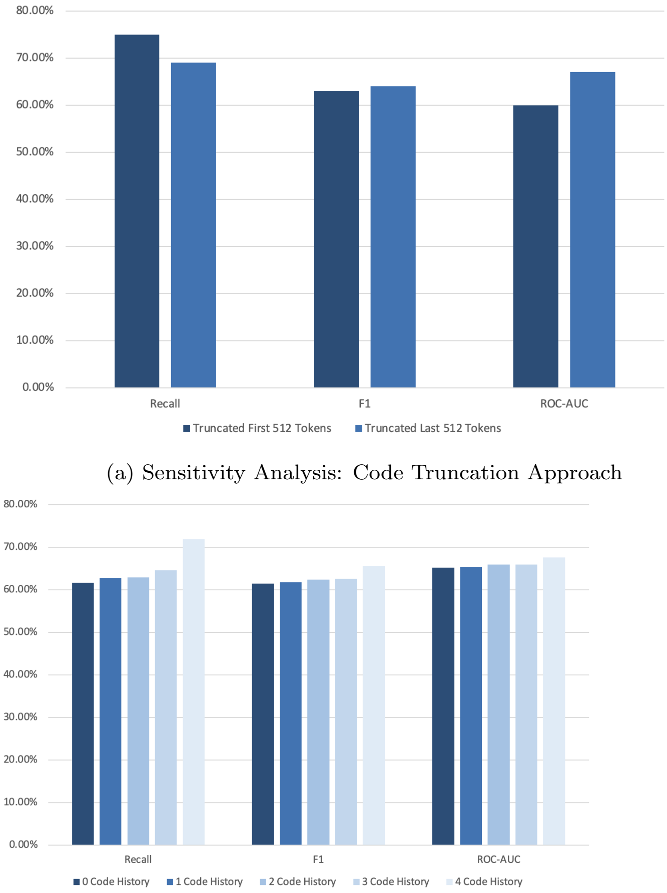
APPENDIX

A. SENSITIVITY ANALYSIS

In this section, we analyze the sensitivity of the model to code truncation and the length of snapshot history.

A.1 Code Truncation

As explained in the Embedding layer section, code truncation is needed for some samples to accommodate to CodeBert's (and Bert's) limitation of 512 tokens. This is important for the BlueJ dataset since it contains 57.4% samples over 512 tokens, compared to the CodeWorkout dataset which contains only 1.6% samples over 512 tokens. In this analysis, we compare truncation to the first 512 tokens vs truncation to the last 512 tokens. Figure 5a presents the result of a sensitivity analysis on the BlueJ dataset. As seen in the figure, the result of truncating the first 512 tokens and the last 512 tokens are similar with a slight advantage to the first 512 tokens truncation. Thus, we decided to truncate the code to the first 512 tokens.



(a) Sensitivity Analysis: Code Truncation Approach

(b) Sensitivity Analysis: Code History Length

Figure 5: Sensitivity Analysis

A.2 Code History Length

Figure 5b presents an analysis of the model's performance when manipulating the number of preceding code snapshots included. Specifically, we change the number of such snapshots from 0 to 4 and present the Recall, F1, and ROC-AUC results for these manipulations on the CodeWorkout dataset. As seen in the figure, the model results improve in all metrics as we add more preceding snapshots. Even though the ROC-AUC metric improves only by 2.42% when we move from zero snapshots to 4 snapshots, the recall metric that represents how well the model detected struggling students improve by 10.2%. This suggests not only that history helps to predict bug-fix-time, but that adding more history may improve the performance of the model, pending on the availability of such data.

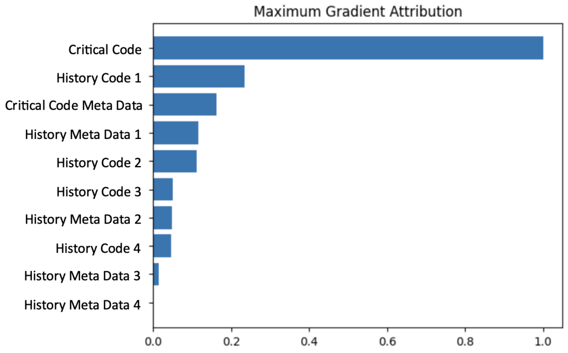


Figure 6: Feature Importance

B. FEATURE IMPORTANCE

To evaluate feature importance, we used Integrated Gradients [28] to calculate feature attribution for the proposed model on the BlueJ dataset. Integrated Gradients are an explainability technique for deep neural networks that visualizes the input feature importance by computing the gradient of the model's prediction output to its input features. In this analysis, we were specifically interested in comparing the importance of different snapshots (latest vs earliest) and the importance of code vs metadata information. To this end, we computed the maximal integrated gradient value for each snapshot and for each metadata group. Calculating the maximum value of each feature group indicates the strongest attribution generated by the group on the output result.

Figure 6 presents the normalized 10 top max attributions. As can be seen in the figure, the critical code snapshot holds the strongest importance, followed by the code snapshot which precedes the critical snapshot (History Code 1). These are then followed in importance by the metadata information from the Critical and History 1 code submissions. Of lower importance are the older code snapshots (History Code 2 and History Code 3). The metadata of Code 2 and Code 3 snapshots and the information of the oldest snapshot (Snapshot 4) are last in line. These results indicate the value captured by both the code itself and the additional metadata information, as well as the value of the information captured from all available historical snapshots (although decreasing as we get earlier in time).

Tool Usage and Efficiency in an Online Test

Gyanesh Jain
Playpower Labs, India
gyanesh.jain@playpowerlabs.com

Aditya Sharma
Playpower Labs, India
aditya@playpowerlabs.com

Nirmal Patel
Playpower Labs, India
nirmal@playpowerlabs.com

Amit.A.Nanavati
Ahmedabad University, India
amit.nanavati@ahduni.edu.in

ABSTRACT

In this study, we analyze data from the National Assessment of Education Progress (NAEP) digital test to understand how digital tool usage relates to the efficiency of answering questions. Digital testing software provides students with on-screen tools such as calculators and scratchpads. We found that students who used digital tools in NAEP were slower in solving the problems but more accurate when the question demanded using the tool. We also found that when students used the tool when it was not needed, they were more likely to be incorrect. Overall, our findings suggest that students need to be trained on how to use the tools and when to use them to make the most use of their testing time.

Keywords

Digital Assessments, Process Data, Student Behavior

1. INTRODUCTION AND PRIOR WORK

The study focuses on the relationship between on-screen tool usage and the efficiency of students. The software used for online testing typically provides tools such as scratchpads and calculators to the students to help them think, and solve problems. Students' usage patterns of these tools can help us model their behavior and help instructors to help students identify tool usage. Data from the NAEP (National Assessment of Educational Progress) used by the US Govt, was used. Computer-based tests are now standard in large-scale assessments.

To ensure the digital competency of the test taker [1, 4] students are subjected to lab-based studies where they are recorded while interacting with the assessment interface [2]. NAEP test provides some tools for the students to aid in problem-solving. Students are provided with quick training before the test starts. Two key tools in the NAEP test are Calculator and Scratchpad. All students in the NAEP test have access to a physical calculator (they are given one if they don't bring their own), and students can also request pencils and scratch paper if needed. The NAEP UI (shown in Figure 1) also has a digital Calculator and a digital Scratchpad that substitute for their physical versions.

Efficient test-takers show distinct digital patterns in the log data. Sahin used Latent Profile Analysis to look at how students allocated times to different test items in the NAEP [5]. They discovered four distinct groups in their sample that they described as 1) little time on the first (problem) visit, 2) balanced time (across problem visits and revisits), 3) little revisit with more time in the end, and 4) little revisit with less time in the end. The researchers found that these four groups differed in the average outcomes, with group 4 scoring the lowest (pg. 21, ibid). Another recent study [3] showed that in the NAEP test, students were more likely to use the digital calculator on calculation-heavy items. When it was used in an ideal way, students were also likely to score higher.

We analyzed the process data from the 2018 NAEP test and compared the students who did not use digital on-screen tools with the ones who did. All students in the NAEP test had access to physical tools, so we wanted to understand student preferences in using digital tools. In our analysis, we compared the time taken by students who used digital tools with those who did not.

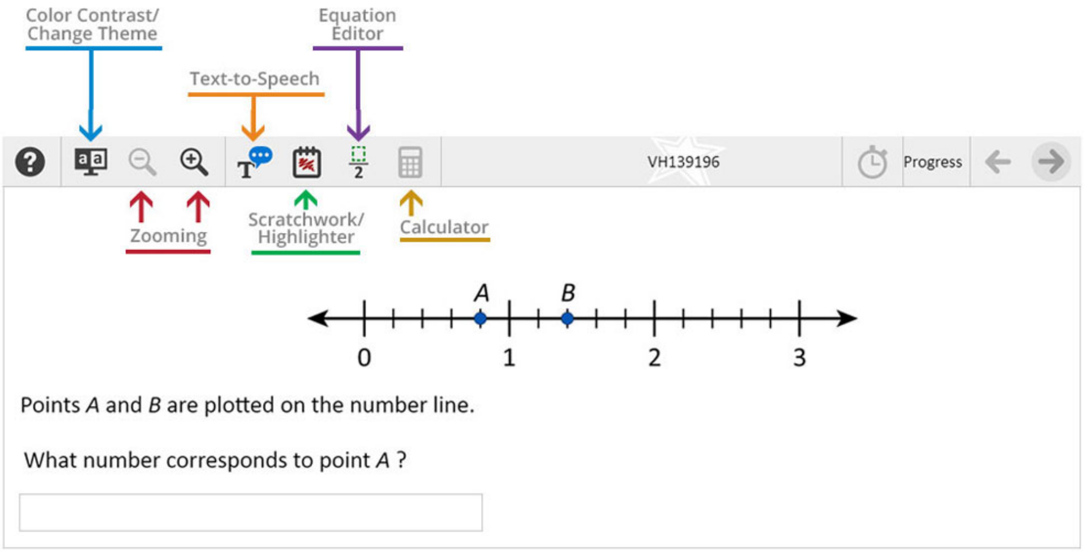
2. DATA

We used a random sample from the 2018 NAEP Mathematics test for Grade 8. Our sample had data N = 1642 students. The digital Math assessment consists of two 30-minute blocks, and our sample had data from the first block. There were a total of 20 questions in the first block. The entire test-taking process of the students was captured by collecting data points for each interaction event. Each student interaction in the digital assessment system resulted in one observation in the dataset. Each observation had seven different variables. They are listed in Table 1 below.

There were forty-two unique types of actions (Observable column in the data). These actions were further coded by us into six different categories: Answer (responding to the item), Navigation (switching between items), Timer (looking at the remaining time), Calculator (using the digital calculator), Scratchpad (using the digital scratchpad), Equation Editor (using the equation editor), and Readability (adjusting the readability of the on-screen text). We consid-

G. Jain, A. Sharma, N. Patel, and A. A. Nanavati. Tool usage and efficiency in an online test. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 406–412, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115735



Points *A* and *B* are plotted on the number line.

What number corresponds to point *A* ?

Figure 1: User interface of the NAEP digital test.

Variable	Description
STUDENTID	Unique identifier of the student
Block	Block of the NAEP test, A or B
AccessionNumber	Unique identifier of the question that the student is attempting
ItemType	Type of the question e.g. MCQ, Fill in the Blank, etc.
Observable	The name of the action that student took e.g. clicking, dragging, scrolling, typing, opening a calculator
ExtendedInfo	Metadata of the student action
EventTime	Time when the student interaction occurred

Table 1: Columns of the NAEP Process Data.

ered Calculator, Scratchpad, and Equation Editor as Digital Tools. We calculated the time students spent doing each type of action, and the combined time students spent using Digital Tools on a question, used to categorize them into tool users and non-tool users. The non-tool users had no tool used for a given question, while the tool users had one or more events related to the digital tools. Once the students were categorized for their tool usage for each question, we compared the two groups for each question, in how quickly they responded to the question items.

We had accuracy data available for eight multiple-choice questions, calculated by looking at the latest option clicked by the students and comparing it with the answer key. Further, the accuracy data was used to compare students who did and did not use the digital tools.

3. RESEARCH QUESTIONS AND METHOD

Our objective was to understand how students who used the on-screen tools differed from those who did not. We wanted to know whether students who used the digital tools were faster at responding to the test items. We also wanted to understand whether the tool usage behavior was different across items and if there were any differences in the proficiency of the students using the tools versus not using the tools.

- **RQ1**: What are the differences in item response times for the students who use the digital tools versus those who do not?
- **RQ2**: Are the tool usage preferences similar or different across question items?
- **RQ3**: Are there any differences in the scores of students who can identify tool usage correctly compared to those who don't?

For RQ1, we used the t-Test to compare the item response times of the groups of students who used the digital tools and those who did not. For RQ2, for each question, we compared the number of students who used digital tools for answering it with those who did not use the tools. Given that we had a random sample from the test, we expected that the student preference seen in our data would generalize to the target population. To answer RQ3, we used correctness data from the multiple choice questions of the test and calculated what proportion of the students using the tools were getting the items correct.

4. RESULTS

RQ1: *We found that, on average, students who used the digital tools took more time to respond to the items than those who did not use the tools.* Figure 2 summarizes our findings for each question. We can see that for some questions, the difference in the response time between non-tool and tool users is as much as double. We performed mean comparisons for items where we had correctness data for the correct, incorrect, and not attempted results. Appendix A contains the results of the t-Tests, where we see that most of the

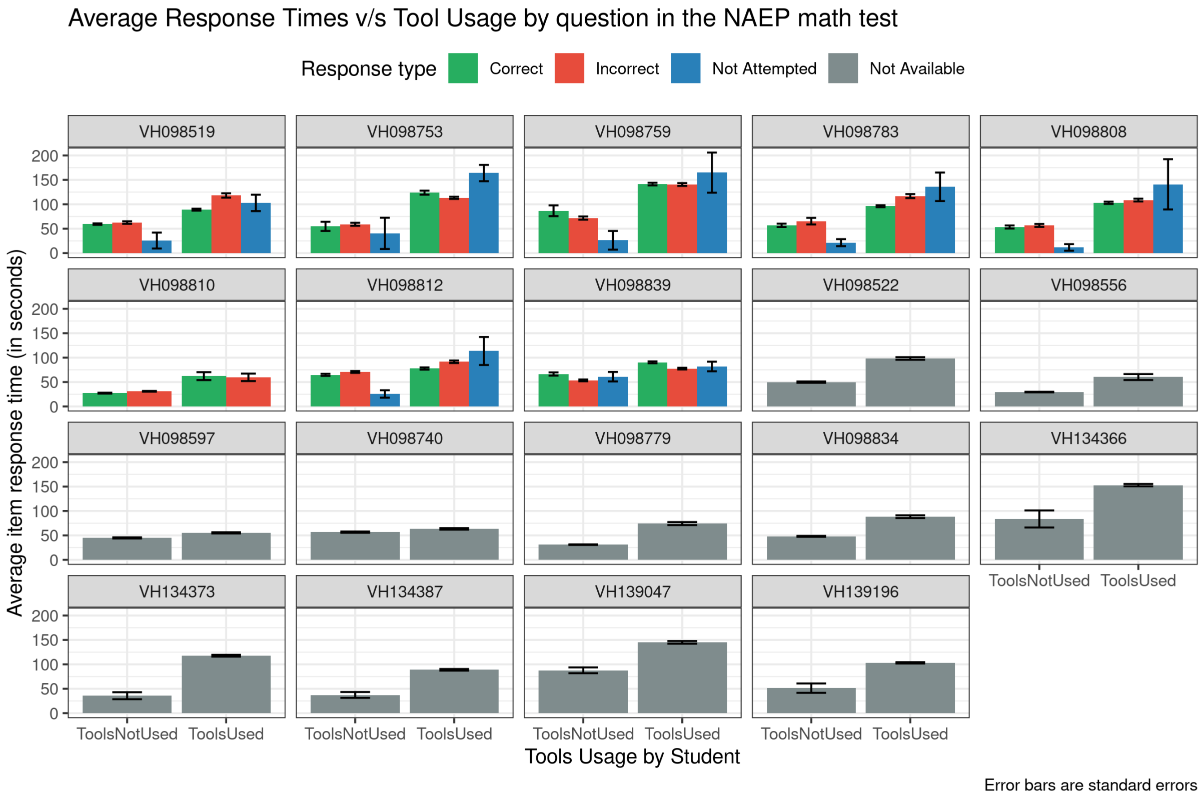


Figure 2: Comparison of item response times for students who used the digital tools and who did not use the digital tools. The answer key was only available for multiple-choice questions.

differences were statistically significant. This answers RQ1 and tells us that students who used the on-screen tools were slower than the ones who did not use the on-screen tools. We do not have precise data about what the non-tool users used. Maybe they did some guesswork (given the multiple-choice questions), or they used physical tools. Based on the content of the questions (provided in Appendix C), we can say that mental math may not be sufficient to solve most of the questions.

RQ2: *We found that students preferred to use tools in some questions and not use them in others.* This is consistent with findings from [3], where they observed that calculator use was more in calculation-heavy items. Question 810, a recall question (shown in Appendix C), did not need a calculator at all, and we can see that out of the 1642 students, 1573 (95.8%) did not use any tool while answering this question. For Questions 753, 759, 783, and 808, more than 80% of the students preferred to use the tools (whether correct or incorrect). For Questions 812 and 519, the tool users were 49.5% and 56.4%, telling us that students did not clearly prefer to use tools for these questions. Looking at Appendix C, we can see that Question 812 may not require tool usage. In summary, Table 2 answers RQ2 and shows us that in some questions, tools were preferred, in some, they were not, and in others, there was mixed behavior.

RQ3: *Overall, we found that the tool-using group of students scored more on average than the non-tool-using group.* We

Question	Tool Usage	Correct (N)	Incorrect (N)	Prop. Correct
VH098519	ToolsNotUsed	246	462	65.25
VH098519	ToolsUsed	251	665	72.60
VH098753	ToolsNotUsed	221	29	11.60
VH098753	ToolsUsed	1000	328	24.70
VH098759	ToolsNotUsed	261	18	6.45
VH098759	ToolsUsed	741	604	44.91
VH098783	ToolsNotUsed	130	161	55.33
VH098783	ToolsUsed	259	1049	80.20
VH098808	ToolsNotUsed	146	173	54.23
VH098808	ToolsUsed	653	657	50.15
VH098810	ToolsNotUsed	697	876	55.69
VH098810	ToolsUsed	39	30	43.48
VH098812	ToolsNotUsed	483	318	39.70
VH098812	ToolsUsed	461	324	41.27
VH098839	ToolsNotUsed	322	79	19.70
VH098839	ToolsUsed	637	438	40.74

Table 2: Number of correct and incorrect students by their digital tool usage. We can see that for some questions, the total number of students who used the tools is higher than the ones who did not use the digital tools. We can also see that tool users scored more on average for some questions. In questions where tools were not required, the tool users scored less.

can see in Table 2, students typically scored more in questions where tools were used. Appendix B shows where the students scored significantly more when using digital tools. The biggest difference was seen in Question 759, a question on calculating averages. Here, 44.9% of the tool users answered correctly, whereas only 6.5% of the non-tool users answered correctly. We do not know why so many students who did not use the tools got the question incorrect because, as per NAEP policy, they all had access to physical calculators. It may be possible that students' digital tool use is an indicator of their other abilities. We know that students who answered without using the digital tools answered the question faster, though some of those responses could be guesses.

5. DISCUSSION

Based on the results, we can see that the students who did not use the tools provided in the NAEP digital interface were faster in answering the question - whether correct or incorrect. Since we did not have data on students' outside activities, we cannot say anything about why the students not using the digital tools were faster. We also tried to sequence modelling on students, but the patterns were too complex for the scope of this poster (Appendix C).

For certain Questions tool usage was not required (based on the content of the question). These questions could be solved without tools, and it was seen that the proportion of correct and incorrect responses for non-tool users is statistically insignificant (Appendix B). For certain Questions which required calculation, we still found that the difference between correct and incorrect non-tool users was insignificant, which is interesting and should be investigated further. In all questions, students have to decide whether to use the tool, implying that student training needs to be conducted on not just how to use the tool but also *when* to use the tool. As the difference between incorrect and correct proportions when not using a tool is positive, it implies that many students were either hesitant to use the tools, inept at using them, or found the tools lacking.

Our analysis did not utilize the fine-grained process data provided by the NAEP system. The step-by-step data of student actions can show how they utilized the tools. To help students be more efficient while taking the test, we can provide personalized feedback based on their usage patterns. For the students who answered correctly, we can find tool usage patterns that were more efficient than theirs (if available) and provide the closest and fastest patterns as suggestions. This can guide the students in avoiding unnecessary steps while solving the problem and making the most of their time. It may be worthwhile to consider having practice tests where the digital test-taking interface disables access to tools for the questions when they are unnecessary. Appropriate tool usage can help students save time and have fewer digital distractions during the test.

6. CONCLUSION AND FUTURE WORK

Our study found that students taking the NAEP test differed in their on-screen tool usage behavior. Students who used the digital tools were typically slower in responding to the items than those who did not. If the question item demanded tool use, then students preferred to use the tools and also scored higher when they used the tools. Our findings show that when taking digital tests, students are better off if they know when to use the tools and when not to use them. A future study can analyze the nuanced processes of tool usage and compare efficient and inefficient digital tool usage. The process data can provide students with personalized recommendations on how to use the tools more efficiently and save time while taking the test. We could also look at the sequence of tool usage and non tool usage amongst students, attempting NAEP.

7. REFERENCES

[1] Antonio Calvani, Antonio Fini, Maria Ranieri, et al. Digital competence in k-12: theoretical models, assessment tools and empirical research. *Anàlisi: quaderns de comunicació i cultura*, pages 157–171, 2010.

[2] Richard P Durán, Ting Zhang, David Sañosa, and Fran Stancavage. Effects of visual representations and associated interactive features on student performance on national assessment of educational progress (naep) pilot science scenario-based tasks. *American Institutes for Research*, 2020.

[3] Yang Jiang, Gabrielle A Cayton-Hodges, Leslie Nabors Oláh, and Ilona Minchuk. Using sequence mining to study students' calculator use, problem solving, and mathematics achievement in the national assessment of educational progress (naep). *Computers & Education*, 193:104680, 2023.

[4] Fanny Pettersson. On the issues of digital competence in educational contexts–a review of literature. *Education and information technologies*, 23(3):1005–1021, 2018.

[5] Füsun Şahin. Exploring the relations between students' time management strategies and test performance. *Paper presented at the Annual meeting of the National Council for Measurement in Education*, 2019.

APPENDIX

A. RESPONSE TIME COMPARISON

i) **For Correct Answers (T-value calculated by Time consumed by tool not used- tool not used)**

AccessionNumber	**VH098519**	**VH098753**	**VH098759**	**VH098783**	**VH098808**	**VH098810**	**VH098812**	**VH098839**
T-Value	-11.759	-6.733	-4.811	-9.962	-12.677	-4.275	-4.114	-6.251
P-value	<2.2e-16	<2.2e-16	<2.2e-16	<2.2e-16	<2.2e-16	<2.2e-16	<2.2e-16	<2.2e-16

ii) **For Incorrect Answers (T-value calculated by Time consumed by tool not used- tool not used)**

AccessionNumber	**VH098519**	**VH098753**	**VH098759**	**VH098783**	**VH098808**	**VH098810**	**VH098812**	**VH098839**
T-Value	-10.689	-14.246	-15.657	-6.507	-12.565	-3.711	-6.49	-9.497
P-value	<2.2e-16	<2.2e-16	<2.2e-16	<2.2e-16	<2.2e-16	<2.2e-16	<2.2e-16	<2.2e-16

iii) **For Unattempted Answers (T-value calculated by Time consumed by tool not used- tool not used)**

AccessionNumber	**VH098519**	**VH098753**	**VH098759**	**VH098783**	**VH098808**	**VH098812**	**VH098839**
T-Value	-3.295	-3.423	-3.057	-3.802	-2.49	-2.965	-1.514
P-value	0.0046	0.0012	0.0121	0.0012	0.0472	0.0069	0.1355

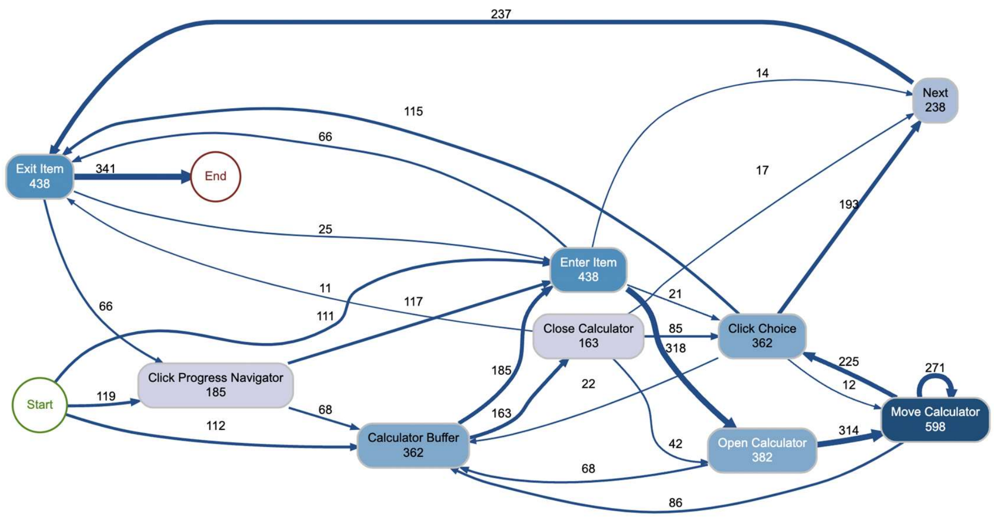
B. CORRECTNESS PROPORTION FOR NON-TOOL USERS

Question No.	**Correct When Tool Not Used**	**Incorrect When Tool Not Used**	**Difference**	**Count Correct**	**Count Incorrect**	**Proportion**	**z-test**	**p-value**	**Sig**
VH098519	0.410	0.495	0.085	1127	497	0.436	3.185	0.0025	Yes

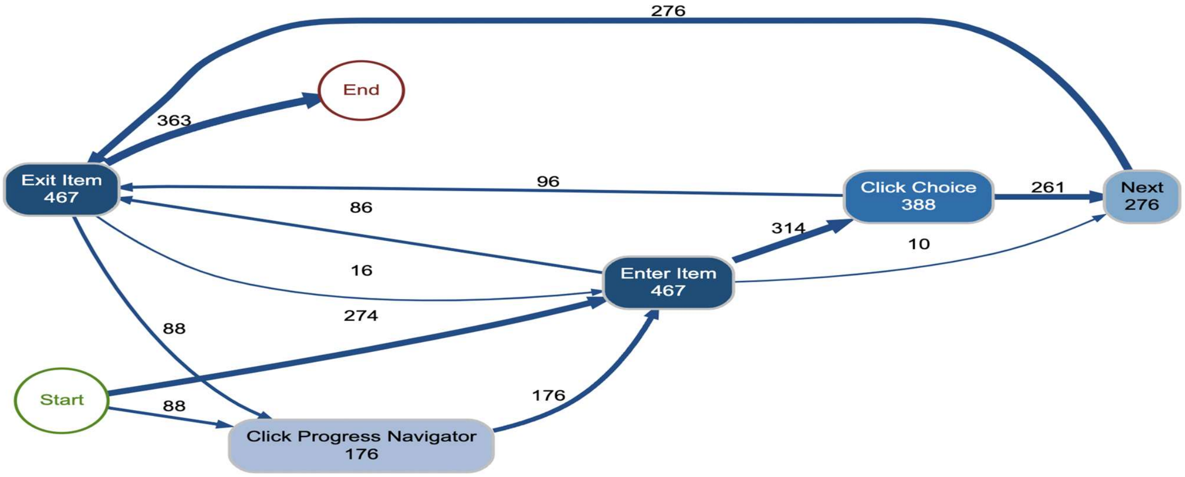
VH098753	0.081	0.181	0.100	357	1221	0.158	4.541	0.0000	Yes
VH098759	0.029	0.260	0.232	622	1002	0.172	12.025	0.0000	Yes
VH098783	0.133	0.334	0.201	1210	389	0.182	8.944	0.0000	Yes
VH098808	0.208	0.183	-0.026	830	799	0.196	-1.307	0.1698	No
VH098810	0.967	0.947	-0.020	906	736	0.958	-1.996	0.0544	No
VH098812	0.495	0.512	0.016	642	944	0.505	0.638	0.3254	No
VH098839	0.153	0.336	0.183	517	959	0.272	7.538	<2.2e-16	Yes

C. SEQUENCE MODELLING

i) Where Calculator was used



ii) Where no tool was used



D) Link for the NAEP test:-

https://www.nationsreportcard.gov/nqt/

A Conceptual Model for End-to-End Causal Discovery in Knowledge Tracing

Nischal Ashok Kumar, Wanyong Feng, Jaewook Lee, Hunter McNichols,
Aritra Ghosh, Andrew Lan
University of Massachusetts Amherst
{nashokkumar,wanyongfeng,jaewooklee,wmcnichols,arighosh,andrewlan}@umass.edu

ABSTRACT

In this paper, we take a preliminary step towards solving the problem of causal discovery in knowledge tracing, i.e., finding the underlying causal relationship among different skills from real-world student response data. This problem is important since it can potentially help us understand the causal relationship between different skills without extensive A/B testing, which can potentially help educators to design better curricula according to skill prerequisite information. Specifically, we propose a conceptual solution, a novel causal gated recurrent unit (GRU) module in a modified deep knowledge tracing model, which uses i) a learnable permutation matrix for causal ordering among skills and ii) an optionally learnable lower-triangular matrix for causal structure among skills. We also detail how to learn the model parameters in an end-to-end, differentiable way. Our solution placed among the top entries in Task 3 of the NeurIPS 2022 Challenge on Causal Insights for Learning Paths in Education. We detail preliminary experiments as evaluated on the challenge's public leaderboard since the ground truth causal structure has not been publicly released, making detailed local evaluation impossible.

Keywords

Causal Discovery, Knowledge Tracing, Response Data

1. INTRODUCTION

Knowledge Tracing (KT) [1] refers to the problem of estimating a student's understanding or mastery of certain skills, concepts, or knowledge components through their responses to questions and using these estimates to predict future performance. KT methods are frequently utilized in modern online education platforms to determine the knowledge levels of many students to enable the platform to provide personalized feedback and recommendations, ultimately leading to better learning results [19]. KT methods are limited in how they represent the relationship between skills; One key limitation is that most do not model the **causal** relationships between skills. Most KT methods simply treat human expert-provided skill tags as a flat structure (with a few exceptions, such as [27], that organize skills hierarchically as trees). As a result, these models are not capable of providing meaningful pedagogical insights, i.e., predicting future student performance if a particular instructional plan is applied instead of the actual plan applied.

Causal analysis tools are a perfect fit to address these limitations in KT. The task of *causal discovery*, i.e., learning causal relationships among different skills from observational data, is especially important. First, it helps educators learn prerequisite relationships among skills. This can guide educators in ordering topics within their curriculum, and can guide students to review prerequisite information when they are stuck on a question [2]. Second, causal relationships among skills helps us with the task of *causal inference*, i.e., estimating the effect of a particular pedagogical treatment or intervention. Traditionally, these tasks are addressed through randomized controlled trials which are difficult to scale. Therefore, incorporating causal discovery into KT methods has the potential to become a scalable alternative since it can be done solely from observational student response data. Performing causal discovery directly from observational student response data is challenging since it is not straightforward to estimate treatment effects from observational data with incomplete or no knowledge of the causal relationship between skills. This problem is referred to as the *end-to-end causal inference* problem, where we discover the causal graph and estimate treatment effects together.

1.1 Contributions

In this paper, we take a **preliminary** step towards learning causal ordering among skills from student response data. This task is proposed in the NeurIPS 2022 Challenge on Causal Insights for Learning Paths in Education[1]. Our proposed **conceptual** solution is, to the best of our knowledge, the first KT method to learn the causal structure among human expert-provided skill tags directly from observational data in an *end-to-end* manner. Specifically, our contributions in this paper are as follows:

- First, we propose an interpretable *causal structure* model that characterizes both i) the dependency among skills using a lower-triangular matrix and ii) their prerequisite ordering using a permutation matrix.

N. A. Kumar, W. Feng, J. Lee, H. McNichols, A. Ghosh, and A. Lan. A conceptual model for end-to-end causal discovery in knowledge tracing. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 413–418, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115727

[1] https://eedi.com/projects/neurips-2022

We hypothesize that this module can be combined with any existing KT method that rely on human expert-provided skill tags.

- Second, as a (among the top) solution[2] to Task 3 in the NeurIPS 2022 Challenge, we apply our causal structure module to a variant of deep knowledge tracing (DKT) [17], with a *causal* gated recurrent unit (GRU) module at its core, due to i) the simple nature of DKT and ii) its good empirical performance in our experiments.
- Third, we detail our experimental results based on the public leaderboard of the NeurIPS 2022 Challenge. We are honest up front that our evaluation is limited since i) the ground-truth causal structure data is not publicly released and ii) the nature of this brand new task means that there are no baselines to compare against.

2. RELATED WORK

2.1 KT methods

Existing KT methods can be classified along several different axes, the first of which is how they represent the student knowledge representation variable h. Classic Bayesian KT methods, such as those in [9, 15, 28], treat student knowledge as a latent binary variable. Recent methods like deep learning-based KT methods, such as [4, 14, 17, 22, 29], treat student knowledge as hidden states in neural networks. This setup results in models that excel at predicting future performance but have limited interpretability [3]. Another major axis is how KT methods represent responses, questions, skills, and time steps. To represent student responses, most existing KT methods treat them as binary-valued indicating response correctness. However, a few methods, such as option tracing [5] and predict partial analysis [26], have characterized student responses as non-binary-valued by analyzing the specific options selected on multiple-choice questions. Another exception is [11], which uses large language models to predict open-ended student responses in a generative way. To represent questions and skills, most existing KT methods one-hot encode them based on question IDs or skill tags [24], except [11, 12]. To represent time steps, most existing KT methods treat each question as a discrete time step, with a few exceptions such as [25], which considers the exact, continuous time elapsed between responses.

2.2 Causal Analysis Methods

In the field of education, there exist very few works on causality and especially few in the context of KT. [8] is closely related to our work, where the authors study the relationship between courses in higher educational institutions using historical student performance data. They use matching methods and regression to determine the average treatment effect (ATE). Along similar lines, [20] and [21] developed theory and methods for analyzing A/B testing data and presented studied data collected from real-world randomized controlled trials. These works focus on causal inference, i.e., assuming that the structure is given and the focus is on estimating the treatment effect. However, the data we use from Eedi contains fine-grained skills, defined as the smallest elements of learning among primary/middle school students. Therefore, our work is different from these works in terms of both the goal and the educational context. The only existing works that study causal discovery in the context of KT are [10] and [13]. The former uses a special model structure that has some similarity to ours to model knowledge state transitions among uninterpretable latent skills. The authors showed that their method, while simple, is highly accurate in predicting unobserved student responses, but do not evaluate on whether the identified causal structure is valid. Our proposed method to learn latent causal structure is closely based on the structural equation model (SEM) [16]. SEM enables us to estimate the relationships between observed and latent variables, offering valuable insights into their underlying relationships. The hypothesized causal relationship among variables is represented as a directed acylic graph (DAG). In this work, our goal is to learn the causal structure graph $\mathcal{G}$.

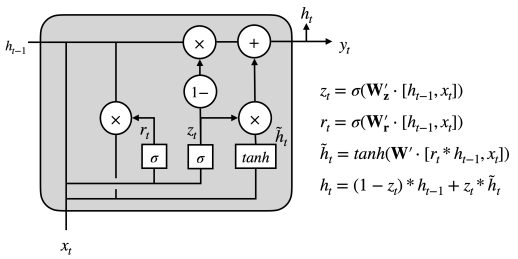

h_{t-1}
h_t
y_t
$\times$
$+$
$1-$
$\times$
$\times$
r_t
z_t
$\tilde{h}_t$
σ
σ
$tanh$
x_t
$z_t = \sigma(\mathbf{W}'_{\mathbf{z}} \cdot [h_{t-1}, x_t])$
$r_t = \sigma(\mathbf{W}'_{\mathbf{r}} \cdot [h_{t-1}, x_t])$
$\tilde{h}_t = tanh(\mathbf{W}' \cdot [r_t * h_{t-1}, x_t])$
$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$

Figure 1: The implementation of a causal GRU cell. All the GRU weight matrices, $\mathbf{W}_z$, $\mathbf{W}_r$, and $\mathbf{W}$ are multiplied by the causal mask $\mathbf{M} = \mathbf{PLP}^T$, resulting in $\mathbf{W}'_z$, $\mathbf{W}'_r$, and $\mathbf{W}'$.

3. METHODOLOGY

We now detail our conceptual causal KT method.

3.1 Basic Setup

The basic KT model contains two components:

$$\mathbf{h}_{j,t} \sim f(\mathbf{h}_{j,t-1}, \mathbf{x}_{j,t}). \tag{1}$$

$$p(Y_{j,t} \mid \mathbf{h}_{j,t}, i_{j,t}). \tag{2}$$

For a student j at time step t, The knowledge estimation component in Eq. (1) estimates the current knowledge state $\mathbf{h}_{j,t}$ given the previous knowledge state $\mathbf{h}_{j,t-1}$ and the student's performance on the problem $\mathbf{x}_{j,t}$ as inputs. The response prediction component in Eq. (2) outputs the prediction of the student's likelihood of answering the next question $Y_{j,t}$ correctly given the current knowledge state $\mathbf{h}_{j,t}$ and the next question index $i_{j,t}$ as input. During the learning process, the KT model needs to maximize the predicted likelihood across responses of all students, i.e., $\sum_j \sum_t \log p(Y_{j,t} \mid Y_{j,1}, \ldots, Y_{j,t-1})$.

We adopt the DKT setup detailed in [18] for consistency. Since incorporating causal learning into the base KT model introduces additional parameters, we use the gated recurrent unit (GRU) as the transition model instead of long short-term memory (LSTM) for computational efficiency. For response prediction, we simply use a single linear layer over the hidden states of the GRU. For causal discovery, i.e., learning the causal structure among skills, we use the causal GRU module detailed below.

[2] The code for our solution can be found at: `https://github.com/umass-ml4ed/Neurips-Challenge-22`

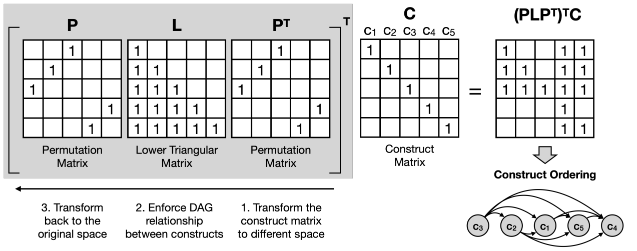

P
L
P^T
T
C
C1 C2 C3 C4 C5
(PLP^T)^TC
=
Permutation Matrix
Lower Triangular Matrix
Permutation Matrix
Construct Matrix
Construct Ordering
3. Transform back to the original space
2. Enforce DAG relationship between constructs
1. Transform the construct matrix to different space
C3
C2
C1
C5
C4

Figure 2: The intuition behind causal GRU. Here, $\mathbf{M} = \mathbf{PLP}^T$. $M_{i,j} = 1$ if and only if $[\mathbf{h}_{t-1}]_j$ influences $[\mathbf{h}_t]_i$ where $\mathbf{h}_t$ is the student's knowledge state at time-step t.

3.2 Causal GRU

We now detail the structure of the Causal GRU. From now on, we drop the student index j for notation simplicity. We define a permuted causal mask $\mathbf{M}$ that represents the causal ordering and structure between skills. The $\mathbf{L}$ matrix represents the *causal structure/skill dependency*, and the $\mathbf{P}$ matrix represents the *causal ordering*. The permuted causal mask $\mathbf{M}$ is calculated in Eq. (3) as first multiplying $\mathbf{P}$ by $\mathbf{L}$ to obtain the updated causal structure and multiplying by $\mathbf{P}^T$ to transform the causal structure into the original space.

The parameters in the Causal GRU are masked, i.e., element-wise multiplied by the permuted causal mask $\mathbf{M}$ in Eq. (4). By masking out some parameters, we zero out parameters that do not satisfy the causal graph. This step ensures that there is no relationship between the hidden states of the non-causally dependent skills in latent student knowledge states. The latest student knowledge state estimation $\mathbf{h}_t$ is calculated in Eq. (5). The input $\mathbf{x}_t$ is represented as a one-hot vector with the dimension size equals to the number of skills C. The entry of value ±1 represents whether the student can correctly answer the question corresponding to the skill. The input $\mathbf{h}_{t-1}$ is the previous student knowledge state estimation. The implementation detail of a Causal GRU cell can be found in Fig. 1.

$$\mathbf{M} = \mathbf{PLP}^T, \tag{3}$$

$$\mathbf{W}^{'} = \mathbf{M} \odot \mathbf{W}, \tag{4}$$

$$\mathbf{h}_t = GRU_c(\mathbf{h}_{t-1}, \mathbf{x}_t). \tag{5}$$

3.2.1 Causal Ordering

One important element of the causal GRU is the *causal ordering* matrix $\mathbf{P}$, which we set to be a permutation matrix. By definition, a permutation matrix has exactly one entry of 1 in each row and each column and 0s elsewhere. Since multiplying a matrix by a permutation matrix permutes the order of the columns/rows of that matrix, the permutation matrix is naturally capable of sorting skills into order based on prerequisite relationships. However, the binary and discrete nature of the permutation matrix makes the learning process non-differentiable. To solve this problem, we introduce a relaxed version of the problem by approximating a permutation matrix with a doubly stochastic matrix, i.e., one where all entries are non-negative and the summation of each column/row is equal to 1, i.e., $P_{i,k} \in [0,1]$, $\sum_k P_{i,k} = 1 \, \forall i$, $\sum_i P_{i,k} = 1 \, \forall k$. Instead of learning a doubly stochastic matrix directly, which is very difficult, we learn a matrix of free parameters $\bar{\mathbf{P}}$, from which we can obtain $\mathbf{P}$ after applying the *Sinkhorn* operator [23] $\mathbf{P} = \text{Sinkhorn}(\bar{\mathbf{P}})$.

The *Sinkhorn* operator works as follows: First, starting with the base matrix $\bar{\mathbf{P}}$, we subtract the largest entry of the matrix from each entry, multiply each entry with the temperature hyper-parameter, and pass it through an exponential function. Second, we apply a series of row and column normalizations by dividing each entry of the column/row by the summation of all the entries in the column/row. In our implementation of the *Sinkhorn* operator, there are two hyper-parameters: *temperature* and *unroll*. The temperature hyper-parameter specifies the extent of the continuous relaxation: the larger the temperature hyper-parameter, the closer $\mathbf{P}$'s entries are to either 0 or 1. The unroll hyper-parameter specifies the number of times row/column normalization is carried out: the more times the normalization is applied, the closer $\mathbf{P}$ is to satisfying the row/column normalization constraints.

3.2.2 Causal Structure

The other important element of the causal GRU is the *causal structure/ skill dependency* matrix $\mathbf{L}$, which we set to be lower triangular. By definition, a lower triangular matrix is one in which all the elements above the principal diagonal of the matrix are 0. This matrix is important since it specifies the causal structure among the skills. Once the skills are ordered using the *causal ordering* matrix $\mathbf{P}$, we apply the *causal structure* matrix $\mathbf{L}$ to regularize student knowledge state transitions across time steps. Due to its lower-diagonal structure, an entry $L_{i,k} > 0$ with $i > k$ implies that skill k is a prerequisite of skill i. Therefore, since the causal GRU weight matrices are masked by the $\mathbf{PLP^T}$ matrix, at the next time step t, the entry in the latent student knowledge vector that corresponds to skill i, $[\mathbf{h}_t]_i$, depends only on the entries in the previous knowledge state that correspond to prerequisites of skill i, i.e., $[\mathbf{h}_{t-1}]_k \, \forall k$ s.t. $L_{i,k} > 0$.

The L matrix being lower diagonal ensures that the resulting causal structure is a DAG. This means that if skill C_1 depends on C_2 then C_2 cannot depend on C_1. As a concrete example, Fig. 2 visualizes the effect of applying a mask $\mathbf{M} = \mathbf{PLP}^T$ on the skill matrix C. The skill matrix C represents 5 skills where each skill is represented as a one-hot vector. The causal ordering matrix $\mathbf{P}$ is applied to the skill matrix to give a skill ordering of C_3, C_2, C_1, C_5, C_4 which is in the order of decreasing pre-requisites (C_3 being the most pre-requisite of all skills). We further apply the $\mathbf{L}$ matrix (in this case a lower diagonal matrix with all ones) that specifies that every subsequent skill depends on the preceding one. For example, it specifies that C_4 depends on all of C_1, C_2, C_3, and C_5; C_5 depends on C_1, C_2, and C_3 and so on.

One easy choice for $\mathbf{L}$ is to set its lower-diagonal part to be all ones; this setting means that every subsequent skill causally depends on all the previous skills. However, in practice, causal dependencies among skills nay not be this dense; most skills will only be causally related to a few other skills. To resolve this problem, we can make the $\mathbf{L}$ matrix learnable by restricting the lower diagonal elements to be either 0 or

Table 1: Results on different model variants.

Model Variant	Leaderboard F_1 Score
No Embedding No Adaptive	0.11
No Embedding Adapative	0.17
Embedding (300D) Adaptive	0.33
Embedding (300D) Adaptive Learnable L	0.43

1. We do this by learning a matrix of free parameters $\bar{\mathbf{L}}$, from which we can obtain $\mathbf{L}$ after applying the element-wise sigmoid operator $\mathbf{L} = \text{sigmoid}(\alpha\bar{\mathbf{L}})$. A large value of the temperature parameter $\alpha > 0$ will push entries to be close to either 0 or 1 but not in between.

3.2.3 *Skill Embeddings*

We use a learnable dense embedding to represent each skill and alter both the input and the output layers of the causal GRU. We learn an embedding matrix $\mathbf{E}$ where each column $\mathbf{e}_c$ represents the embedding of skill c. We treat the dimension of $\mathbf{e}_c$ as a hyperparameter. For the input layer, we use another learnable embedding $\mathbf{d}$, which is either added or subtracted from the skill embedding depending on the correctness of the previous answer. We then learn the input to the causal GRU using $NN(\mathbf{e}_c \pm \mathbf{d})$ where NN is a single-layer neural network. For the output, we use $p(Y_t) \sim NN_o([\mathbf{e}_c^T, \tilde{\mathbf{h}}_t^T]^T)$, where $\tilde{\mathbf{h}}_t$ is a masked version of $\mathbf{h}_t$ with the only non-zero entry being the one that corresponds to the skill of the next question that we are predicting. Here NN_o is a single-layer neural network that predicts the probability of the correct answer.

4. EXPERIMENTS

4.1 Data and Challenge Description

We participated in Task 3 of the NeurIPS Challenge cohosted by Eedi, Microsoft Research, and Rice University [6]. The goal of this task is to discover the causal relationships between different skills, or *constructs* (as defined by Eedi, which means the smallest unit of learning; for example, "mental addition and subtraction" is a construct within the main topic "math"), and evaluate the effect of learning one *skill* on another. Questions in this dataset are multiple-choice, with a single correct option and three distractors that are designed to assess a single skill. The challenge hypothesis is that it is possible to discover the hidden relationship behind different skills through analyzing the responses to a large number of diagnostic questions. The challenge uses an F_1 score-based metric which calculates the similarity between the predicted adjacency matrix $\hat{A}$ and the true adjacency matrix A.

4.2 Model Learning and Hyperparameters

The dataset consists of 1855 skills and 6468 students. We set the default skill embedding dimension to 300. We use an adaptive strategy and start with small values of the temperature and unroll and linearly increase their values over a set of epochs. We set the initial temperature and unroll to 2 and 5 respectively and linearly increase the values with a factor of 2 and 5 respectively for every 10 epochs. We train the model for 50 epochs with a batch size of 64, and a learning rate of 5e-4 using four Nvidia Tesla 2080 GPUs with a GPU memory of 12GB each which takes about 6 hours. After training, to obtain the final *causal structure* matrix $\mathbf{L}$ we apply a post-processing step. We define a hyperparameter κ such that all values of the $\mathbf{L}$ matrix less than κ are set to 0 and all values greater than or equal to κ are set to 1.

4.3 Results and Discussion

In Table 1, we show the results of different model variants. We report the leaderboard F_1 score obtained in our experiments. We see that the F_1 score is 0.11 for the case where we are not using the skill embeddings. Using an adaptive strategy increases the F_1 score by 0.06, which suggests that the adaptive strategy is helpful during model training. We also report the results corresponding to the skill embeddings and the learnable $\mathbf{L}$. We see that using an embedding dimension of 300 almost doubles the F_1 score. This observation confirms our hypothesis that using skill embeddings increases the representational capacity of the neural network model and hence performs better. When the causal structure matrix $\mathbf{L}$ is learnable, we see that we get a further 0.1 increase in the F_1 score. The increase in the F_1 score on using the learnable $\mathbf{L}$ configuration of the model shows that it is better to learn the explicit causal dependence of skills instead of assuming a dense representation where each skill depends on all the skills preceding it.

5. CONCLUSIONS AND FUTURE WORK

In this work, we proposed a conceptual method for learning causal structure among skills from student response data, as a part of our solution to the NeurIPS 2022 Challenge on Causal Insights for Learning Paths in Education. Our method is a novel causal knowledge tracing method that enables us to learn the causal structure in an end-to-end manner while performing knowledge tracing. Unfortunately, due to space limitations, we cannot show a qualitative example of the learned causal structure among skills. We believe that our work should inspire future works in the direction of building causal knowledge tracing methods on observational student response data. First, it is important to evaluate the accuracy of the the learned causal structure between skills, either against human domain experts or via A/B testing. Second, it is important to apply our causal module to more flexible knowledge tracing methods, such as attention-based methods, to see whether it is applicable and effective. Third, it is important to develop ways to leverage both the opinion of human experts and our data-driven causal discovery model, in a human-in-the-loop manner. The former may be less accurate but the latter requires extensive training data; a hybrid human-AI collaboration may be able to take the best from both sides.

6. ACKNOWLEDGEMENTS

The authors thank the NSF (under grants 1917713, 2118706, 2202506, 2215193) for partially supporting this work. We also thank the organizers of the 2022 NeurIPS Challenge on Causal Insights for Learning Paths in Education for proposing a new and meaningful task.

7. REFERENCES

[1] A. Corbett and J. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User-adapted Interact.*, 4(4):253–278, Dec. 1994.

[2] M. C. Desmarais. Mapping question items to skills with non-negative matrix factorization. *ACM SIGKDD Explorations Newsletter*, 13(2):30–36, 2012.

[3] T. Gervet, K. Koedinger, J. Schneider, T. Mitchell, et al. When is deep learning the best approach to knowledge tracing? *Journal of Educational Data Mining*, 12(3):31–54, 2020.

[4] A. Ghosh, N. Heffernan, and A. S. Lan. Context-aware attentive knowledge tracing. In *Proc. ACM SIGKDD*, pages 2330–2339, 2020.

[5] A. Ghosh, J. Raspat, and A. Lan. Option tracing: Beyond correctness analysis in knowledge tracing. In *Int. Conf. Artif. Intell. Educ.*, pages 137–149. Springer, 2021.

[6] W. Gong, D. Smith, Z. Wang, C. Barton, S. Woodhead, N. Pawlowski, J. Jennings, and C. Zhang. Instructions and guide: Causal insights for learning paths in education. *arXiv preprint arXiv:2208.12610*, 2022.

[7] A. Hyvärinen, K. Zhang, S. Shimizu, and P. O. Hoyer. Estimation of a structural vector autoregression model using non-gaussianity. *Journal of Machine Learning Research*, 11(5), 2010.

[8] P. Kaur, A. Polyzou, and G. Karypis. Causal inference in higher education: Building better curriculums. In *Proceedings of the Sixth (2019) ACM Conference on Learning@ Scale*, pages 1–4, 2019.

[9] M. Khajah, Y. Huang, J. González-Brenes, M. Mozer, and P. Brusilovsky. Integrating knowledge tracing and item response theory: A tale of two frameworks. In *Proc. Int. Workshop Personalization Approaches Learn. Environ.*, volume 1181, pages 7–15, 2014.

[10] A. S. Lan, C. Studer, and R. G. Baraniuk. Time-varying learning and content analytics via sparse factor analysis. In *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 452–461, Aug. 2014.

[11] N. Liu, Z. Wang, R. Baraniuk, and A. Lan. Open-ended knowledge tracing for computer science education. In *Conference on Empirical Methods in Natural Language Processing*, pages 3849–3862, 2022.

[12] Q. Liu, Z. Huang, Y. Yin, E. Chen, H. Xiong, Y. Su, and G. Hu. Ekt: Exercise-aware knowledge tracing for student performance prediction. *IEEE Trans. Knowl. Data Eng.*, 33(1):100–115, 2019.

[13] S. Minn, J.-J. Vie, K. Takeuchi, H. Kashima, and F. Zhu. Interpretable knowledge tracing: Simple and efficient student modeling with causal relations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12810–12818, 2022.

[14] S. Pandey and J. Srivastava. Rkt: Relation-aware self-attention for knowledge tracing. *arXiv preprint arXiv:2008.12736*, 2020.

[15] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a Bayesian networks implementation of knowledge tracing. In *Proc. Int. Conf. User Model. Adaptation Personalization*, pages 255–266, 2010.

[16] J. Pearl. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009.

[17] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Proc. NeurIPS*, pages 505–513, 2015.

[18] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. *Advances in neural information processing systems*, 28, 2015.

[19] S. Ritter, J. R. Anderson, K. R. Koedinger, and A. Corbett. Cognitive tutor: Applied research in mathematics education. *Psychonomic Bulletin & Review*, 14(2):249–255, 2007.

[20] A. Sales, A. Botelho, T. Patikorn, and N. T. Heffernan. Using big data to sharpen design-based inference in a/b tests. In *Proceedings of the Eleventh International Conference on Educational Data Mining*, 2018.

[21] A. C. Sales and J. F. Pane. Student log-data from a randomized evaluation of educational technology: A causal case study. *Journal of Research on Educational Effectiveness*, 14(1):241–269, 2021.

[22] D. Shin, Y. Shim, H. Yu, S. Lee, B. Kim, and Y. Choi. Saint+: Integrating temporal features for ednet correctness prediction. In *11th Int. Learn. Analytics Knowl. Conf.*, pages 490–496, 2021.

[23] R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964.

[24] S. Sonkar, A. E. Waters, A. S. Lan, P. J. Grimaldi, and R. G. Baraniuk. qdkt: Question-centric deep knowledge tracing. In *Proceedings of the International Conference on Educational Data Mining*, 2020.

[25] C. Wang, W. Ma, M. Zhang, C. Lv, F. Wan, H. Lin, T. Tang, Y. Liu, and S. Ma. Temporal cross-effects in knowledge tracing. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 517–525, 2021.

[26] Y. Wang and N. Heffernan. Extending knowledge tracing to allow partial credit: Using continuous versus binary nodes. In *Int. conf. artif. intell. educ.*, pages 181–188. Springer, 2013.

[27] Y. Yang, J. Shen, Y. Qu, Y. Liu, K. Wang, Y. Zhu, W. Zhang, and Y. Yu. Gikt: A graph-based interaction model for knowledge tracing. In *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2020.

[28] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In *Int. Conf. artif. intell. educ.*, pages 171–180. Springer, 2013.

[29] J. Zhang, X. Shi, I. King, and D.-Y. Yeung. Dynamic key-value memory networks for knowledge tracing. In *Proc. Int. Conf. World Wide Web*, pages 765–774, Apr. 2017.

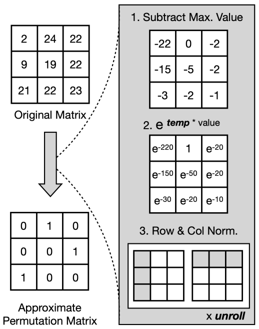


Figure 3: The *Sinkhorn* operator is a smooth operator that outputs an approximate permutation matrix.

Table 2: Results on varying the cutoff for the L matrix, κ.

κ	F_1 Score
0.42	0.43
0.45	0.43
0.435	0.43
0.48	0.43
0.495	0.43
0.51	0.33
0.525	0.18

APPENDIX

A. STRUCTURAL EQUATION MODELING

In SEM, given a collection of random variables $\mathbf{x} = (x_1, \cdots, x_D)$ and a causal directed acyclic graph (DAG) $\mathcal{G}$, each variable x_i is generated using its parents $Pa(i;\mathcal{G})$ in the DAG and an exogenous noise variable ϵ_i: $x_i = f_i(\{x_j\}_{j \in Pa(i;\mathcal{G})}) + \epsilon_i$. The structural vector autoregression (SVAR) model extends SEM to random variables that form a time series $\mathbf{x}_t = (x_{1,t}, \cdots, x_{D,t})$ for time steps $t \in \{1, \cdots, T\}$ [7]. The influence of one variable on others can be instantaneous or lagged behind for a few time steps. The SEM model is given by

$$x_{i,t} = \sum_{\tau=0}^{k} f_{i,\tau}(\{x_{j,t-\tau}\}_{j \in Pa(i,t,\tau;\mathcal{G})}) + \epsilon_{i,t},$$

where $Pa(i,t,\tau;\mathcal{G})$ are random variables from time step $t - \tau$ that influence random variable $x_{i,t}$. One can also use a single latent state $\mathbf{h}\cdot, t$ to model the influence of past random variables. The SEM becomes

$$\mathbf{h}_{i,t} = f_i(\{h_{j,t-1}\}_{j \in Pa(i;\mathcal{G})}) + \epsilon_i.$$

B. SINKHORN OPERATION

Fig. 3 shows the working of the Sinkhorn operator.

C. ADDITIONAL RESULTS

We report the results obtained using different hyperparameters of the learnable **L** model configuration. In Table 2, we show the results across different κ values. We vary the values from 0.42 to 0.525 and observe that using any values out of this range gives a leaderboard F_1 score of 0. Among the experimented values for κ we see that we obtain a maximum F_1 score of 0.43 for all values in the range of 0.42 to 0.495. The maximum F_1 score for κ values in the range of 0.42 to 0.495 means that using very large or very less values of κ does not give the optimal skill dependency.

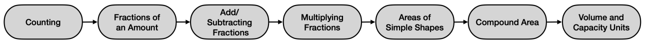


Figure 4: An example of learned causal ordering among skills in actual student response data.

We perform qualitiative analysis of our proposed method. Fig. 4 shows an example from the DAG obtained from the learned adjacency matrix for causal relations. Here, we represent the constructs based on their subject names, and an arrow from subject i to subject j implies that subject i is a pre-requisite of subject j. Here, in the figure, we can see that "Counting" is the most pre-requisite skill. The subsequent skills in fractions depend on "Counting". Calculating areas of simple figures depends on fraction multiplication. In the same way, calculating the volume depends on calculating the area. Hence, from this, we can see that we are able to learn a meaningful DAG using our methodology.

LECTOR: An attention-based model to quantify e-book lecture slides and topics relationships

Erwin D. Lopez Z.
Kyushu University
lopez.zapata.erwin.242@s.kyushu-u.ac.jp

Tsubasa Minematsu
Kyushu University
minematsu@limu.ait.kyushu-u.ac.jp

Yuta Taniguchi
Kyushu University
yuta.taniguchi.y.t@gmail.com

Fumiya Okubo
Kyushu University
fokubo@ait.kyushu-u.ac.jp

Atsushi Shimada
Kyushu University
atsushi@limu.ait.kyushu-u.ac.jp

ABSTRACT

The use of digital lecture slides in e-book platforms allows the analysis of students' reading behavior. Previous works have made important contributions to this task, but they have focused on students' interactions without considering the content they read. The present work complements these works by designing a model able to quantify the e-book LECture slides and TOpic Relationships (LECTOR). Our results show that LECTOR performs better in extracting important information from lecture slides and suggest that readers' topic preferences extracted by our model are important factors that can explain students' academic performance.

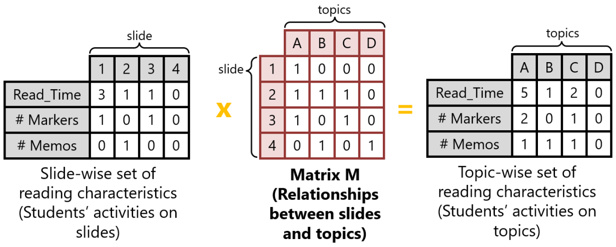
Keywords

e-book, reading behavior, keyphrase extraction, multimodal learning analytics

1. INTRODUCTION

The adoption of e-learning technologies in blended courses can help instructors better understand students' learning behaviors and make more informed revisions of lessons and materials [9]. Examples of these technologies include the e-book reading systems used in university classrooms to distribute lecture materials. By modeling students' interactions on these systems, instructors can analyze their reading behavior and support their learning process [14, 22, 15].

Several works have investigated how to model e-book reading users based on their set of reading characteristics [1, 34, 24, 8, 2]. Nevertheless, their models did not consider the content that students read [31], information that may be important for improving the course content's structures [16], or providing process-oriented feedback to students [27].



	1	2	3	4
Read_Time	3	1	1	0
# Markers	1	0	1	0
# Memos	0	1	0	0

Slide-wise set of reading characteristics (Students' activities on slides)

	A	B	C	D
1	1	0	0	0
2	1	1	1	0
3	1	0	1	0
4	0	1	0	1

Matrix M (Relationships between slides and topics)

	A	B	C	D
Read_Time	5	1	2	0
# Markers	2	0	1	0
# Memos	1	1	1	0

Topic-wise set of reading characteristics (Students' activities on topics)

Figure 1: Topic-wise data generation

Since lecture slide data consists of text and images, their integration into current models poses several challenges to be addressed [7, 31]. Both text and image processing are difficult tasks that recent advances in computer science are attempting to address in different domains. Furthermore, considering multimodal data would require formulating a model able of integrating the different data sources.

In this context, the present work takes the first step by focusing on the text-processing task. We propose the model LECTOR, which uses Natural Language Processing (NLP) techniques to estimate a quantitative relationship between a lecture slide and a topic. By performing this estimation, we can convert a slide-wise set of reading characteristics into a topic-wise set of reading characteristics (Figure 1). Accordingly, we validate LECTOR's performance on this task against previous models.

2. RELATED WORK

2.1 Text processing in e-book lecture slides

Previous studies describe the use of e-book lecture slide text to address various problems, such as slide summarization [28], personalized recommendation [21, 23], and learning footprint transfer [33]. Almost all of these works used the TF-IDF method [26] to process their slides [28, 33, 21]. Other works use hierarchical models to perform this process [32, 5], but they require human labeling of all the text in the slides [3], a task that can be burdensome for teachers.

In addition, a previous study estimated topic reading time from e-book user data by considering only the slides where

E. D. L. Z., T. Minematsu, Y. Taniguchi, F. Okubo, and A. Shimada. Lector: An attention-based model to quantify e-book lecture slides and topics relationships. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 419–425, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115729

the topic was written [31]. We can reformulate this method as a matrix product (Figure 1), where they assigned a relationship of 1 when the topic appears in a given slide, and 0 in other cases (referred to as “Binary score” in this paper).

2.2 Keyphrase extraction from documents

Our problem is reduced to an unsupervised keyphrase extraction task if we consider lecture slides as documents and topics as key phrases. The state-of-the-art studies on this task use pre-trained models (e.g., Doc2Vec [18], ELMo [25], BERT [12]) to represent words as embedding vectors [6, 29, 13]. Then, their methods estimate the similarity between key phrases and documents from the cosine similarity of their corresponding embedding representations [6, 29, 13].

3. PROPOSED MODEL

LECTOR extracts a set of topic candidates from all the slides of a given course and assigns a single score to each slide-topic pair (Figure 2). This score is defined as a linear combination of two different scores, one based on the words' importance and the other on the similarity between the topic and the slide embeddings.

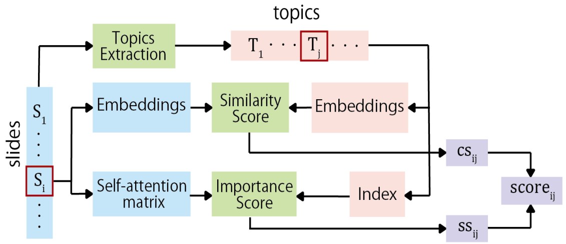


Figure 2: Overview of our proposed model.

3.1 Topics extraction

We consider a topic to be an observable entity (keyphrase). Models such as EmbedRank [6] and AttentionRank [13] use the Part-Of-Speech to generate noun phrases that become their possible key phrases. In our case, we work with slides written in Japanese and use the Bi-LSTM-based NLP library Nagisa to identify the nouns. Then, we define single nouns and n-gram sequences (n=2) of nouns as our topics.

3.2 Word embeddings and attention matrix

We use a BERT model (fine-tuned on all the course slides' text in the MLM task [12]) to estimate a self-attention matrix A^i and a set of word embeddings E^i for each slide. We then correct these token-wise values to word-wise values [10].

3.3 LECTOR's importance score

For a given slide s_i, we quantify the attention a_{ij} that words w belonging to a given topic t_j receive from all the other words w within the slide s_i by summing the different weights of the matrix A^i as shown in Equation 1.

$$a_{ij} = \sum_{w \in t_j} \sum_{w' \in s_i \setminus \{w\}} A^i_{w'w} \quad (1)$$

Since this score is strongly influenced by the frequency of the topic's words f_j, the importance score (ss_{ij}) is calculated by considering the Smooth Inverse Frequency [4] (Equation 2).

$$ss_{ij} = a_{ij} \left(\frac{k}{k + f_j} \right) \quad (2)$$

3.4 LECTOR's similarity score

For a given slide s_i, we estimate its embedding representation P^i_s as a weighted average of its corresponding word embeddings E^i (Equation 3).

$$P^i_s = \sum_{w \in s_i} Weight\,(w)\, E^i_w \quad (3)$$

We define the word weight as the probability of belonging to the discourse of the given slide. We consider that this discourse is given by a general discourse introduced in the first slide of the lecture material and a specific discourse introduced by the title of the respective slide (Figure 3).

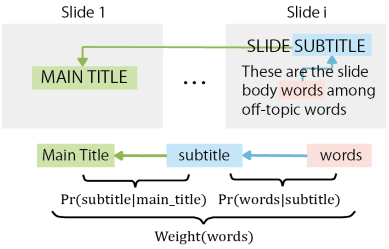


Figure 3: Overview of the weight calculation process.

Accordingly, given the set of title and body embeddings E^i_{st} and E^i_{sb}, the Weights are calculated as shown in Equation 4. In Appendix A, we detail the formulation and estimation of these Weights from the set of word embeddings.

$$Weight = Pr\,(w_t \in st_i | st_1)\, Pr\,(w_t \in sb_i | st_i) \quad (4)$$

Finally, the similarity score is given by the cosine similarity between the topic t_j and slide s_i embeddings [6, 29, 13].

$$b_{ij} = \frac{P^i_s \cdot E^j_t}{||P^i_s||\;||E^j_t||} \quad (5)$$

$$cs_{ij} = \left(\frac{1}{f_j} \sum_{topic j} b_{ij} \right) f^{\alpha}_j, \alpha \in [0, 0.25] \quad (6)$$

3.5 LECTOR's final score

The final score for a given topic t_j and slide s_i is a linear combination of the previously normalized importance and similarity scores (Equation 7). The parameter d defines the importance of each score value.

$$score_{ij} = d * ss_{ij} + (1 - d) * cs_{ij} \quad (7)$$

LECTOR's final output is the matrix M, whose elements M_{ij} are the final scores between slides s_i and topics t_j.

4. RESULTS AND DISCUSSION

4.1 Dataset

Our dataset consists of the textual content of 620 slides from 22 e-book materials delivered in the course “Programming Theory” in the year 2019 (before the pandemic restrictions). This course was offered by the School of Engineering at Kyushu University for 7 weeks.

4.2 First Experiment formulation

The ground-truth values to evaluate LECTOR's estimates are given by the relationships between different topics and slides. However, to find them empirically, we would need a large number of samples because these relationships are perceived differently by different people. Furthermore, given the large number of topics and slides in a course, we would need millions of ground truth labels for each sample.

For this reason, our experiment is designed to indirectly evaluate the estimates of the models. Similar to works on keyphrase extraction, we assume that the most important topics should have the highest relationships with the course content (the different slides). For a given topic t_j, we define its keyphrase candidate score mt_j as the sum of the scores obtained across all slides (Equation 8).

$$mt_j = \sum_{i=1}^{\#slides} M_{ij} \tag{8}$$

We use the mt_j values to extract the most important topics of the course. Our ground-truth labels are given by the course keywords extracted from the course syllabus (*"Scheme"*, *"Data Structure"*, *"List Processing"*, *"Recursion"*, *"Expression"*, *"Condition"*, *"Design Recipe"*, *"Function"*, *"High-level function"*). We define @n as the set that contains the top n topics according to the scores mt_j. By comparing this set to the ground truth, we can measure the model performance.

We considered three baselines. The first is given by the TF-IDF model [26], which is predominant in the slide text processing literature. The second is given by the AttentionRank model [13], which represents the state-of-the-art in unsupervised keyphrase extraction. The third model is given by the previously described *Binary score* model proposed by [31].

4.3 First Experiment results

Our results are summarized in Table 1. We can see that AttentionRank outperforms all the other models with an F-score of 28.68% when considering the 5 most important topics. This result shows the high performance of this state-of-the-art model even in a different domain (slides unstructured text). This F-score was achieved by identifying 2 keyphrases in its five most important topics. As we can see in Table 2, while all the models identified the keyphrase *"Function"* as the most important topic, AttentionRank also identified the keyword *"Recursion"* as its fourth most important topic. From Table 2, we can also note that despite all the other models achieving the same F-score, the TF-IDF and Binary models are more influenced by the frequency of the topics, estimating topics such as *"i"* and *"define"* as one of their most important ones.

At $n = 10$, we can see that the attention-based models outperform the TF-IDF and Binary models. Specifically, AttentionRank, LECTOR Similarity score, and LECTOR achieve an F-score of 31.68%. At $n = 15$, LECTOR outperforms all the other models with an F-score of 33.44%. We can see the same result when comparing the best F-score obtained by each model and the mean of the results obtained in the first n@100 sets. These results show that AttentionRank has difficulty finding new keyphrases, whereas LECTOR does not.

Table 1: Summary of the F-score results for Experiment 1. The mean is calculated from the first n@100 sets

n	Model	P	R	F1
5	Baseline (TF-IDF)	20.00	11.11	14.39
	Baseline (AttentionRank)	**40.00**	**22.22**	**28.68**
	Baseline (Binary score)	20.00	11.11	14.39
	LECTOR Importance Score	20.00	11.11	14.39
	LECTOR Similarity Score	20.00	11.11	14.39
	LECTOR	20.00	11.11	14.39
10	Baseline (TF-IDF)	10.00	11.11	10.63
	Baseline (AttentionRank)	**30.00**	**33.33**	**31.68**
	Baseline (Binary score)	10.00	11.11	10.63
	LECTOR Importance Score	20.00	22.22	21.15
	LECTOR Similarity Score	**30.00**	**33.33**	**31.68**
	LECTOR	**30.00**	**33.33**	**31.68**
15	Baseline (TF-IDF)	20.00	33.33	25.11
	Baseline (AttentionRank)	20.00	33.33	25.11
	Baseline (Binary score)	20.00	33.33	25.11
	LECTOR Importance Score	20.00	33.33	25.11
	LECTOR Similarity Score	20.00	33.33	25.11
	LECTOR	**26.67**	**44.44**	**33.44**
Best	Baseline (TF-IDF)	20.00	33.33	25.11
	Baseline (AttentionRank)	**37.50**	33.33	35.39
	Baseline (Binary score)	23.08	33.33	27.38
	LECTOR Importance Score	20.00	33.33	25.11
	LECTOR Similarity Score	25.00	**44.44**	32.11
	LECTOR	33.00	**44.44**	**38.20**
Mean	Baseline (TF-IDF)	11.68	46.00	15.53
	Baseline (AttentionRank)	12.63	40.89	15.65
	Baseline (Binary score)	11.26	43.33	14.77
	LECTOR Importance Score	12.68	50.22	16.85
	LECTOR Similarity Score	14.48	59.67	19.69
	LECTOR	**15.19**	**61.56**	**20.70**

In Table 2, we can see that AttentionRank tends to give high scores also to minor topics such as *"define"*, *"else"*, or *"empty"* which may explain its lower performance.

The mentioned problem of AttentionRank has two reasons. The first is that its "Accumulated Self-Attention" is influenced by the word frequencies. In their paper, the authors pointed out that this characteristic can be beneficial in large documents. However, in the context of lecture slides, several words from the domain knowledge of the course can appear repeatedly. For example, the mentioned *"define"* and *"else"* are well used in the program examples of the course "Programming Theory". On the other hand, the design we considered in the LECTOR's importance score limits the influence of the frequency of the words.

However, in the AttentionRank model, topics must also achieve a high "Cross-Attention" value in order to get a high final score. The reason that words like *"define"* and *"else"* are important topics of the model is due to the two discourse hypotheses of AttentionRank. For a given slide, the first assumes that the topic candidate defines the slide discourse, and the second assumes that the slide defines the topic discourse. In the context of noisy and unstructured slide text, this consideration can lead to some problems.

For example, given the topic *"define"* and a slide that contains a programming code example about list processing,

Table 2: Most important topics of each model. ENG: a word originally written in English.

n	TF-IDF	AttentionRank	Binary score	LECTOR
1	function	function	function	function
2	list	example problem	list	data
3	list (ENG)	definition	define (ENG)	list
4	i (ENG)	recursion	definition	definition
5	define (ENG)	example	cond (ENG)	program
6	definition	value	data	computation
7	page	define (ENG)	list (ENG)	function definition
8	data	expression	empty	expression
9	count	argument	count	example problem
10	program	computation	i (ENG)	recursion
11	value	list	value	data definition
12	expression	else (ENG)	expression	list processing
13	cond (ENG)	empty (ENG)	recursion	program design
14	example	element	else (ENG)	recursion function
15	recursion	count	element	exercises

the mentioned model will focus on the context words of *"define"* in the code (including the *"define"* itself) resulting in a high Cross-attention score in this case. Then, when we consider the topics *"list processing"* or *"example code"*, even if the model manages to estimate high scores for these topics, they will be relatively as important as *"define"*.

Similarly, the presence of noise in the slides can highly influence the relative scores, sometimes estimating low scores for a closely related topic and slide pair. In contrast, LECTOR's similarity score considers a singular discourse defined by the main title and slide title that give relatively high scores to topics highly related to this discourse. In the previous example, LECTOR would give higher scores to *"list processing"* and *"example code"* rather than *"define"*, and also would give a higher score to *"define"* rather than a random noise word.

4.4 Second Experiment formulation

Previous studies of students' eye-tracking data have concluded that each student has a different preference for learning content [20]. Accordingly, this experiment aims to compare the topic preferences of students with different grades.

We extract their reading time on the different slides (inside and outside of class) and obtain their slide preferences by normalizing the reading time values across the week. Then, we use LECTOR to quantify their Relative Reading Times for the different topics (Topic RRT), as shown in Figure 1. Finally, we group the students according to their grades (A=24, B=6, C=4, D=6, F=10) and compare both their reading time and RRT distributions. We measure the separability of the distributions by using the Fisher Discriminant Ratio (FDR) and statistically validated them with a T-test.

4.5 Second Experiment results

We can see an example of our results in Figure 4. Figure 4a shows the distribution of the reading time of the students with final grades A and B in the second week after the lecture (out-class). Both distributions overlap, so the FDR is 0.0502 and the significance level (p) of the T-test is 0.3302. In Figure 4b we see the same distributions when we consider the relative time spent reading about "Design method". Here,

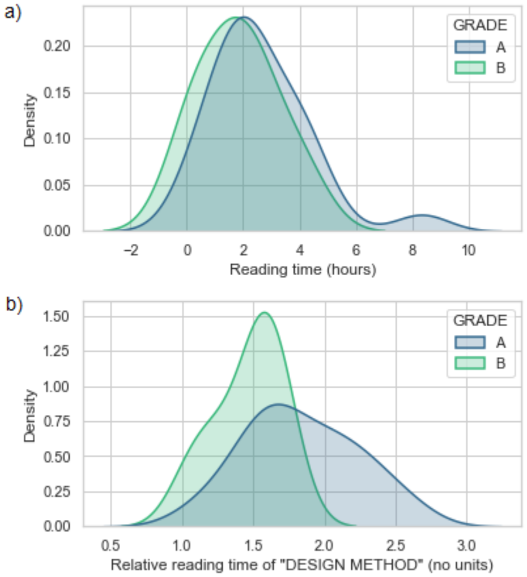


Figure 4: a) Reading time of the students with final grades A and B. b) The same distributions when considering the relative time of reading about the topic *"Design method"*.

students with a final grade of A tend to read more on this topic, resulting in a higher FDR of 5.5802 and a lower p of 0.037 in the T-test.

Our different results are summarized in Table 3. We considered the first 3 weeks of the course because of insufficient data in later weeks due to dropouts. As shown in this table, we have included 5 cases, comparing students with consecutive grades (A-B, B-C, C-D, D-F) and at-risk students (students who failed the course) with non-risk students. The result shown in Figure 4 can be found in the first column and fourth row of the table.

In the results of Reading Time, we can see that students from different groups tend to read the same amount of time. In the case of at-risk and non-risk student groups, we find

Table 3: Fisher Discriminant Ratio between different groups of students in the first 3 weeks of the course.

		A-B	**B-C**	**C-D**	**D-F**	**At-risk**
WEEK 1 (IN-CLASS)	Reading Time	0.0342	0.2615	2.782	0.0229	**1.111***
	Topic RRT	1.4517	**612.44***	46.861	**3.233***	4.3245
	(Topic)	(expressions)	(data)	(exercises)	(design method)	(execution)
WEEK 1 (OUT-CLASS)	Reading Time	0.0409	0.0023	0.0436	0.0085	0.000
	Topic RRT	3.0031	29.3069	**653.11****	72.649	1.4049
	(Topic)	(auxiliary functions)	(problems)	(program design)	(problems)	(auxiliary functions)
WEEK 2 (IN-CLASS)	Reading Time	1.0128	0.6902	0.1735	0.0021	0.0192
	Topic RRT	6.3908	8.4876	**568.83***	1.7794	1.5921
	(Topic)	(problems)	(boolean value)	(problems)	(program)	(program)
WEEK 2 (OUT-CLASS)	Reading Time	0.0502	0.0855	0.2629	0.0913	**0.325***
	Topic RRT	**5.5802***	29.9718	**241.1***	2.8445	**17.92***
	(Topic)	(design method)	(cond expression)	(data analysis)	(body expression)	(exercise problems)
WEEK 3 (IN-CLASS)	Reading Time	0.0503	0.4597	0.1141	0.0142	0.3367
	Topic RRT	11.8214	8.263	7.998	15.061	5.031
	(Topic)	(exercise problems)	(synthetic data)	(synthetic data)	(sorting)	(examples)
WEEK 3 (OUT-CLASS)	Reading Time	0.0234	0.0008	0.2131	1.4279	0.1951
	Topic RRT	**15.166***	**168.33***	**286.84****	42.266	**43.126***
	(Topic)	(templates)	(element count)	(structure element)	(exercice problems)	(exercice problems)

*p<0.05 **p<0.01

statistically significant differences in out-of-class engagement in the second and third weeks. On the other hand, we find statistically significant differences between different groups almost 40% of the time when we consider the Topic RRT, which means that these preferences are good variables to understand the differences between students with different grades. This suggests that works that attempt to predict at-risk students such as [24, 8] may benefit from the integration of models such as LECTOR to obtain more differentiated features.

We can consider student's reading preferences for further analysis. For example, as mentioned earlier, at-risk students engage less outside of class in the second and third weeks. In Table 3, we also see that they tend to focus more on exercise problems. This is a signal that at-risk students adopt a surface learning approach [17], focusing on the content directly related to the assessments. Thus, previous works [1, 34] that have analyzed the students' reading behavior can use the topic preferences to make better reports.

5. LIMITATIONS

The first limitation is the indirect evaluation of the models' estimates. As previously discussed, collecting labels for a direct evaluation is impractical, but if we limit the number of topics to the most important ones we can collect a limited set of labels to conduct a more direct evaluation.

The second limitation is the size of our dataset. To evaluate the generalizability of our model, we need to consider slides from different courses. In a science course, the slides are less structured and include equations or code. In this case, the robustness of LECTOR plays an important role. In addition, our slides are in Japanese and the generality of our results may be affected by the use of other methods for topic extraction in different languages.

6. CONCLUSIONS

We proposed LECTOR, a new model that adapts state-of-the-art keyphrase extraction models to the domain of lecture slides. From our results, we conclude that LECTOR can quantitatively extract the relationships between topics and e-book lecture slides better than previous models when considering noisy text from scientific lecture slides. LECTOR was able to extract important topics (higher F-score) while avoiding frequent out-of-context topics.

LECTOR's topic-wise representation of e-book reading characteristics provides new insights into the students reading behavior. Specifically, it allows to access the students' preferences for some topics and use them to model more detailed behaviors. Our results show that this new model preserves the differences related to reading preferences that exist between students with different final grades.

These responses validate the benefits of integrating attention-based models like LECTOR into reading behavior models. Accordingly, it allows future works to consider students reading preferences in their models. Also, our model can be used for other text processing tasks, such as slide summarization, content recommendation, etc.

7. ACKNOWLEDGMENTS

This work was supported by JST CREST Grant Number JPMJCR22D1 and JSPS KAKENHI Grant Number JP22H00551, Japan.

8. REFERENCES

[1] G. Akçapinar, M.-R. A. Chen, R. Majumdar, B. Flanagan, and H. Ogata. *Exploring Student Approaches to Learning through Sequence Analysis of Reading Logs*, page 106–111. Association for Computing Machinery, New York, NY, USA, 2020.

[2] G. Akçapinar, M. N. Hasnine, R. Majumdar, B. Flanagan, and H. Ogata. Developing an early-warning system for spotting at-risk students by using ebook interaction logs. *Smart Learning Environments*, 6(1):4, May 2019.

[3] F. N. Al-Aswadi, H. Y. Chan, and K. H. Gan. Automatic ontology construction from text: a review from shallow to deep learning trend. *Artificial Intelligence Review*, 53(6):3901–3928, Aug 2020.

[4] S. Arora, Y. Liang, and T. Ma. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*, 2017.

[5] T. Atapattu, K. Falkner, and N. Falkner. A comprehensive text analysis of lecture slides to generate concept maps. *Computers & Education*, 115:96–113, 2017.

[6] K. Bennani-Smires, C. Musat, A. Hossmann, M. Baeriswyl, and M. Jaggi. Simple unsupervised keyphrase extraction using sentence embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 221–229, Brussels, Belgium, Oct. 2018. Association for Computational Linguistics.

[7] P. Blikstein and M. Worsley. Multimodal learning analytics and education data mining: using computational technologies to measure complex learning tasks. *Journal of Learning Analytics*, 3(2):220–238, Sep. 2016.

[8] C.-H. Chen, S. J. H. Yang, J.-X. Weng, H. Ogata, and C.-Y. Su. Predicting at-risk university students based on their e-book reading behaviours by using machine learning classifiers. *Australasian Journal of Educational Technology*, 37(4):130–144, Jun. 2021.

[9] S. K. Cheung, J. Lam, N.Lau, and C.Shim. Instructional design practices for blended learning. In *2010 International Conference on Computational Intelligence and Software Engineering*, pages 1–4, 2010.

[10] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, Aug. 2019. Association for Computational Linguistics.

[11] Y. Cui, Z. Chen, S. Wei, S. Wang, T. Liu, and G. Hu. Attention-over-attention neural networks for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 593–602, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[13] H. Ding and X. Luo. AttentionRank: Unsupervised keyphrase extraction using self and cross attentions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1919–1928, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.

[14] B. Flanagan and H. Ogata. Integration of learning analytics research and production systems while protecting privacy. In *Workshop Proceedings of the 25th International Conference on Computers in Education*, pages 355–360, 12 2017.

[15] B. Flanagan and H. Ogata. Learning analytics platform in higher education in japan. *Knowledge Management and E-Learning*, 10:469–484, 11 2018.

[16] F. Martin, A. Ritzhaupt, S. Kumar, and K. Budhrani. Award-winning faculty online teaching practices: Course design, assessment and evaluation, and facilitation. *The Internet and Higher Education*, 42:34–43, 2019.

[17] F. Marton and R. Säaljö. On qualitative differences in learning—ii outcome as a function of the learner's conception of the task. *British Journal of Educational Psychology*, 46(2):115–127, 1976.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26, pages 3111–3119. Curran Associates, Inc., 2013.

[19] A. Mnih and G. Hinton. Three new graphical models for statistical language modelling. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 641–648, New York, NY, USA, 2007. Association for Computing Machinery.

[20] S. Mu, M. Cui, X. J. Wang, J. X. Qiao, and D. M. Tang. Learners' attention preferences of information in online learning. *Interactive Technology and Smart Education*, 16(3):186–203, Jan 2019.

[21] K. Nakayama, M. Yamada, A. Shimada, T. Minematsu, and R. ichiro Taniguchi. Learning support system for providing page-wise recommendation in e-textbooks. In K. Graziano, editor, *Proceedings of Society for Information Technology & Teacher Education International Conference 2019*, pages 1078–1085, Las Vegas, NV, United States, March 2019. Association for the Advancement of Computing in Education (AACE).

[22] H. Ogata, M. Oi, K. Mohri, F. Okubo, A. Shimada, M. Yamada, J. Wang, and S. Hirokawa. *Learning analytics for E-book-based educational big data in higher education*, pages 327–350. Springer International Publishing, May 2017.

[23] F. Okubo, T. Shiino, T. Minematsu, Y. Taniguchi, and A. Shimada. Adaptive learning support system based on automatic recommendation of personalized

review materials. *IEEE Transactions on Learning Technologies*, 16(1):92–105, 2023.

[24] F. Okubo, T. Yamashita, A. Shimada, Y. Taniguchi, and K. Shin'ichi. On the prediction of students' quiz score by recurrent neural network. *CEUR Workshop Proceedings*, 2163, 2018.

[25] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[26] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[27] G. Sedrakyan, J. Malmberg, K. Verbert, S. Järvelä, and P. A. Kirschner. Linking learning behavior analytics and learning science concepts: Designing a learning analytics dashboard for feedback to support learning regulation. *Computers in Human Behavior*, 107:105512, 2020.

[28] A. Shimada, F. Okubo, C. Yin, and H. Ogata. Automatic summarization of lecture slides for enhanced student previewtechnical report and user study. *IEEE Transactions on Learning Technologies*, 11(2):165–178, 2018.

[29] Y. Sun, H. Qiu, Y. Zheng, Z. Wang, and C. Zhang. Sifrank: A new baseline for unsupervised keyphrase extraction based on pre-trained language model. *IEEE Access*, 8:10896–10906, 2020.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. ukasz Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc., 2017.

[31] J. Wang, T. Minematsu, Y. Taniguchi, F. Okubo, and A. Shimada. Topic-based representation of learning activities for new learning pattern analytics. In *Proceedings of the 30th International Conference on Computers in Education*, pages 268–378, 12 2022.

[32] Y. Wang and K. Sumiya. Semantic ranking of lecture slides based on conceptual relationship and presentational structure. *Procedia Computer Science*, 1(2):2801–2810, 2010. Proceedings of the 1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010).

[33] C. Yang, B. Flanagan, G. Akcapinar, and H. Ogata. Maintaining reading experience continuity across e-book revisions. *Research and Practice in Technology Enhanced Learning*, 13(1):24, Dec 2018.

[34] C. Yin, M. Yamada, M. Oi, A. Shimada, F. Okubo, K. Kojima, and H. Ogata. Exploring the relationships between reading behavior patterns and learning outcomes based on log data from e-books: A human factor approach. *International Journal of Human–Computer Interaction*, 35(4-5):313–322, 2019.

APPENDIX

A. WORDS' WEIGHTS ESTIMATION

A.1 Preliminary definition

Given a set of words $A = \{w_a^1, w_a^2, ..\}$ and $B = \{w_b^1, w_b^2, ..\}$, we will estimate $Pr\,(w_a \in A|B)$: The probability of each word in A being generated under the discourse (context) of the set of words B.

First, the probability that a given word w_a is generated under a given context word w_b is proportional to the inner product of their word embeddings (Equation 9) [4, 19].

$$Pr\,(w_a|w_b) \propto exp\left(e_a \cdot e_b^T\right) \tag{9}$$

With this equation, we can estimate the probability of each word w_a in the set A to be generated under the single context word w_b, as shown in Equation 10.

$$Pr\,(w_a \in A|w_b) = [k_1 exp\Big(e_a \cdot e_b^T\Big), k_2 exp\left(e_a \cdot e_b^T\right), ..] \tag{10}$$

We assume a common proportional constant ($k_1 = k_2 = ...$). Then, we can represent Equation 10 as the softmax of the matrix product between the set of embeddings $E_a = [e_a^1, e_a^2, ..]$ and the context embedding e_b, as shown in Equation 11 (the parameter φ preserves the influence of the proportional constant). This equation can also be interpreted as the cross-attention between the Query e_b and the Key E_a [30].

$$Pr\,(w_a \in A|w_b) = Softmax\left(\frac{e_b \cdot {E_a}^T}{\varphi\sqrt{d_k}}\right) \tag{11}$$

Finally, we can generalize this equation to the context $B = \{w_b^1, w_b^2, ...\}$ by using the approach "Attention over attention" proposed in the study [11].

$$S = \frac{E_b \cdot {E_a}^T}{\varphi\sqrt{d_k}} \tag{12}$$

$$Pr\,(w_a \in A|B) = AV_{row}\left(SF_{col}\left(S\right)\right) SF_{row}\left(S\right) \tag{13}$$

where AV_{row} means average along the row axis, SF_{col} means softmax along the column axis, and SF_{row} means softmax along the row axis.

A.2 Formulation

Given the set of words embeddings E_i for each slide, we split it into the set of title and body embeddings E_{st}^i and E_{sb}^i. Then, the words' Weights are estimated using Equations 11 and 13 as follows:

$$S = \frac{E_{st}^1 \cdot {E_{st}^i}^T}{\varphi\sqrt{d_k}} \tag{14}$$

$$Pr\,(w_t \in st_i|st_1) = AV_{row}\left(SF_{col}\left(S\right)\right) SF_{row}\left(S\right) \tag{15}$$

$$Pr\,(w_t \in sb_i|st_i) = Softmax\left(\frac{E_{st}^i \cdot {E_{sb}^i}^T}{\varphi\sqrt{d_k}}\right) \tag{16}$$

$$Weight = Pr\,(w_t \in st_i|st_1)\, Pr\,(w_t \in sb_i|st_i) \tag{17}$$

Course Concepts: How Readable Are They for ESL Learners?

Yo Ehara
Tokyo Gakugei University
ehara@u-gakugei.ac.jp

ABSTRACT

Massive open online courses (MOOCs) are online courses for multiple learners with different backgrounds, including English-as-a-second-language (ESL) learners. In a MOOC, course concepts are important for diverse learners to grasp what they can learn in the course and its prerequisite knowledge. Previous studies have explored methods to automatically extract concepts from course videos or identify prerequisite concepts in a course. However, as a concept typically consists of several words, it could be difficult for ESL learners to understand what a concept means if they do not know the words in the concept. For example, for "geospatial data," many of them may need an additional explanation of what "geospatial" means in addition to the explanation of the concept. This paper extensively analyzes the readability of MOOC concepts using an openly-available manually-annotated MOOC-concept dataset on computer science and economics and a vocabulary test result dataset of ESL learners with different English skills. We found that the percentage of concepts for which an ESL learner is likely to know all the words is only 25.8% in computer science. In economics, the value is 56.5%. This implies that ESL learners usually require additional vocabulary explanations to understand MOOC concepts. We also show qualitative analyses and that almost half of the concepts are unreadable to ESL learners.

Keywords

Course Concepts, Readability, Second Language Learners

1. INTRODUCTION

Massive open online courses (MOOCs) are online lecture courses intended for use by a large number of learners with different backgrounds, including English-as-a-second-language (ESL) learners. In MOOCs, students learn numerous knowledge concepts, or course concepts, some of which are taught in the course, whereas others are prerequisites of the course. Course concepts are important because learners "with different backgrounds can grasp the essence of the course" [5]. Previous studies have focused on extracting course concepts automatically from course video recordings [5] or identifying prerequisite concepts of a course [5]. However, MOOC learners also include ESL learners. How much additional effort is required to ensure that ESL learners understand the concepts taught in the course? No previous study has extensively investigated this research question, which we address in this paper.

For example, consider the concept of "big data." ESL learners who are willing to listen to an English course usually know both "big" and "data" because both "big" and "data" are high-frequency words on the general corpus; therefore, they are likely to have been mastered by the learners in their previous English studies. In this case, the teacher only needs to explain what "big data" is. Therefore, the effort to teach this concept to ESL learners is almost the same as that to native English speakers. In contrast, when considering the concept of "geospatial data," it is possible that many ESL learners do not know the meaning of the word "geospatial". "Geospatial" is a specialized word that is rare in general corpora. While native English speakers may only need an explanation of "geospatial data," an ESL learner may need an additional explanation of what "geospatial" means, such as "something related to locations and maps." No previous studies have extensively studied the difference in the effort required to teach a concept to native English speakers and ESL learners.

This study estimates how much additional effort is required when teaching concepts to ESL learners using an openly available manually checked MOOC concept list dataset. Specifically, we estimate which words learners are likely to know the meaning of by using a machine-learning method that takes vocabulary test results and the frequency of the general corpus as features. We experimented with manually annotated concept datasets from online courses in computer science and economics. The experiment showed that 60% of the concepts consisted of two English words, and approximately half of the concepts are not readable to almost all learners in the learner vocabulary dataset that we employed.

2. DATASETS

Unlike academic wordlists and specialized terminology extraction studies and their datasets, MOOC concepts refer to the specific knowledge taught in MOOCs. One of the openly available English course concept datasets manually verified

Y. Ehara. Course concepts: How readable are they for esl learners? In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 426–429, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115742

Table 1: Bigram concepts with largest difference between two words in computer science.

Words	Difference	Mean Prob.
right subtree	0.655	0.215
left subtree	0.642	0.212
block tridiagonal	0.581	0.190

Table 2: Bigram concepts with largest difference between two words in economics.

Words	Difference	Mean Prob.
OCO order	0.621	0.215
rediscounted rate	0.602	0.210
salesforce management	0.596	0.209

is that of [5], which is a collection of concepts taken from eight computer science and economics courses on Coursera, one of the most popular English MOOCs. While larger MOOC concept datasets are available in subsequent studies, namely MOOCCube and MOOCCubeX, they are taken from XuetangX, which mainly consists of Chinese courses. Hence, throughout the paper, we use the dataset by [5].

To answer our research question, we also need a dataset from which we can obtain what kinds of words ESL learners know. Since MOOCs are intended to offer courses for many learners with diverse backgrounds over the Web, the ESL learners of the dataset are also preferred to have been collected on the Web. Few datasets meet this criterion because, in most ESL datasets, ESL learners are classroom students of a school; hence, they are not diverse. One such dataset is [1], in which 100 ESL learners answer 100 vocabulary questions. The learners of this dataset were collected using crowdsourcing; hence, they have more diverse backgrounds than classroom learners.

3. EXPERIMENTS

The dataset of [5] contains eight Coursera computer science and economics courses, including their transcripts. Human annotators manually annotated whether k-grams in the transcripts were course concepts or not. In computer science, in total, the dataset has 4,096 concepts; nearly 60% of them consist of two words (bigrams), 18% one word (unigrams), and 22% three words (trigrams). In economics, in total, the dataset has 3,652 concepts; nearly 66% of them consist of bigrams, 10% unigrams, and 24% trigrams.

We also built a classifier that predicts how likely a word is to be known to a learner. To this end, we used the learner

Table 3: Bigram concepts with smallest difference between two words in computer science.

Words	Difference	Mean Prob.
learning rule	9.94×10e-5	0.645
thread programming	3.74×10e-4	0.468
network traffic	4.38×10e-4	0.614

Table 4: Bigram concepts with smallest difference between two words in economics.

Words	Difference	Mean Prob.
federal agency	3.16×10e-6	0.643
quantitative easing	3.69×10e-5	0.426
domestic cresit	1.04×10e-4	0.659

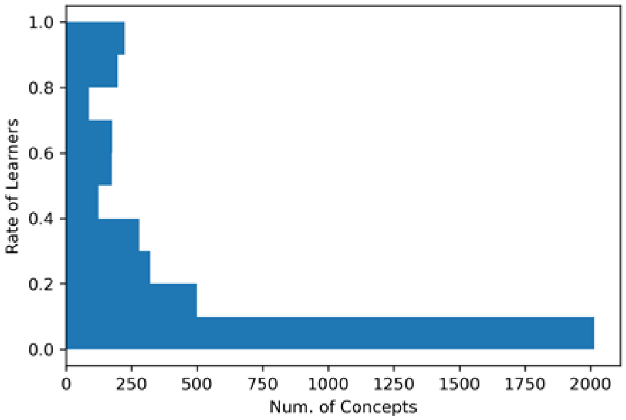


Figure 1: Histogram of Concepts Known to Learners in computer science.

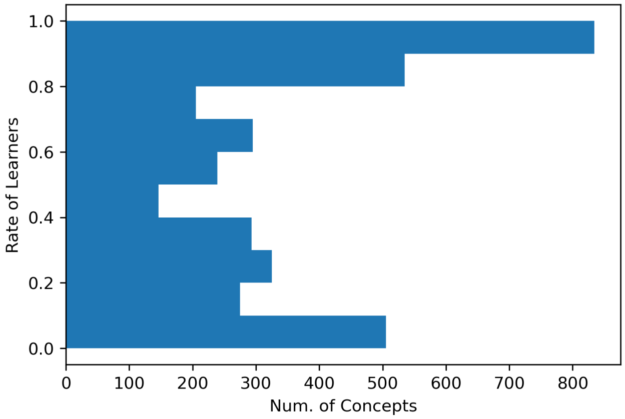


Figure 2: Histogram of Concepts Known to Learners in economics.

vocabulary test dataset of [1]. We built a machine-learning classifier that, given a learner and a word, classifies whether the learner knows the word. Following [1], we used one-hot vectors for learner features and word frequencies taken from the British National Corpus (BNC) and Contemporary Corpus of American English (CoCA) as features; both corpora are general corpora frequently used for teaching ESL learners. In the [1] dataset, the learners' English skills in this dataset are diverse while the test-takers are mainly Japanese because the dataset was built using a Japanese crowdsourcing service called Lancers. The dataset consists of 100 ESL learners; those who do not know even the basic words "computer" and "science" are unlikely to be willing to learn computer science in English, we omitted such lowly skilled learners, resulting in 94 learners. For classification, we used logistic regression because it was previously applied to their dataset and was reported to have high accuracy in measuring ESL learners' readability [3]. For the 10,000 responses (100 learners for 100 vocabulary questions) of the dataset, we first split the data into 9,800 for training data and 200 for test data. The logistic regression was highly accurate as it achieved 86.1% accuracy on the test data in the [1] dataset, whereas the chance rate was 60.3%.

We then applied our classifier to the MOOC concepts. Specifically, for each learner in the dataset, we obtained the probability that the learner knows the word for each word in a concept. By simply taking the product of the probability values of all words in a concept, we obtained the probability that the learner knows the concept: for example, when learner A knows "big" and "data" with probabilities of 0.9 and 0.8, respectively, the probability that learner A knows "big data" is 0.72. Then, if the probability of a concept is equal to or greater than 0.5, we considered that the learner knows the concept.

In computer science, on average, an ESL learner knows 1,058 concepts, which amounts to only 25.8% of the 4,096 concepts, implying that an ESL learner needs an explanation to understand some word(s) in the concept. Figure 1 is a histogram showing what percentage of ESL learners the concepts are known to. We can see that almost half of the concepts are known to less than 10% learners.

In economics, situations are quite different from those in computer science. On average, an ESL learner knows 2,065 concepts, which amounts to only 56.5% of the 3,652 concepts, implying that an ESL learner needs an explanation to understand some word(s) in the concept. Figure 2 is a histogram showing what percentage of ESL learners the concepts are known to. We can see that almost 500 of the concepts are known to less than 10% learners, whereas almost 800 concepts are known to more than 90% learners.

We then focus on the average ESL learner in the dataset and see the concepts that may require special attention when teaching second language learners. To this end, we focus on the bigram concepts and see the difference in the probability known to ESL learners between the two words of which each concept consists. Whereas native-speaker learners know both words and simply need to learn what the concept as a whole means, in addition, ESL learners need to learn what the word in the concept means if the learner does not know the meaning of a word in the concept. What is particularly unintuitive is that one word of the concept is easy for learners, whereas the other(s) is/are not. In this case, the words constituting the concept may seem easy to native-speaker teachers because one of the words is easy. However, as the other word(s) is/are not, such concepts can be confusing to ESL learners. Hence, we list up these words in the following paragraphs.

In computer science, Table 1 shows the bigram concepts with the largest difference in the mean probability known to the average learner between the two words in the concepts, and Table 3 shows the concepts with the smallest difference. We can see that the words particularly difficult for the average learner were "subtree" and "tridiagonal."

In economics, Table 2 shows the bigram concepts with the largest difference in the mean probability known to the average learner between the two words in the concepts, and Table 4 shows the concepts with the smallest difference. We can see that the words particularly difficult for the average learner were "OCO" and "rediscounted."

4. RELATED WORK AND DISCUSSION

In this study, we used concept data from an English MOOC. On the other hand, if the language is not limited to English, a study on MOOCs includes data from a large MOOC in Chinese in [7]. Conceptual information is expensive for teachers to tag, so the study [8] helps teachers by automatically assigning conceptual information. Such research will eventually be used to recommend courses for MOOCs [9].

However, these studies have not paid particular attention to the common case of MOOC participants being second language learners. As for the readability of second language learners, there are mainly two approaches to collecting the dataset for experiments.

One approach is to collect data from language teachers. In this approach, language teachers teaching second language learners read each text in the dataset and label the difficulty. Particularly, the task for automatically assessing the readability of texts is called automatic readability assessment (ARA) and has been studied extensively in [6, 4]. The strength of this approach is that we can easily obtain one gold label for each text. The weakness of this approach is that the quality of the annotations heavily depends on the expertise of the language teachers.

In contrast, another approach is to collect data from language learners themselves. English learners cannot directly annotate what texts are difficult for them. However, unlike the method of having English teachers annotate the texts, this method can obtain information directly from the English learners. Therefore, it is not affected by the noise of what kind of students the English learners have taught in the past. In this approach, data from language learners taking a vocabulary test consisting of short sentences is available to the public [1]. This study also followed this approach. Especially, [2] investigates the readability of scientific abstracts.

5. CONCLUSIONS

To conclude, we made preliminary analyses of the readability of MOOC concepts to ESL learners. Importantly, Figure 1 shows that, for nearly 2,000 concepts of the 4,096 ones, ESL learners also need an explanation of the words used in the concept to understand the explanation of the concept. According to the Figure 2, this situation is relaxed in the field of economy, but still, about 500 concepts out of 3652 concepts, or 13.7%, are not understood by ESL learners.

These results indicate that if ESL learners could know the meaning of the basic words used in the concepts before taking these courses, their understanding of the courses might be greatly improved. To this end, future work includes personalized support systems that automatically explain the words in the concepts.

6. ACKNOWLEDGMENTS

This work was supported by JST ACT-X Grant Number JPMJAX2006, Japan. We used the ABCI infrastructure of AIST. We appreciate the anonymous reviewers' valuable comments.

7. REFERENCES

[1] Y. Ehara. Building an English Vocabulary Knowledge Dataset of Japanese English-as-a-Second-Language Learners Using Crowdsourcing. In *Proc. of LREC*, May 2018.

[2] Y. Ehara. Semantically adjusting word frequency for estimating word difficulty from unbalanced corpora. In *Companion Proc. of LAK*, 2020.

[3] Y. Ehara. No meaning left unlearned: Predicting learners' knowledge of atypical meanings of words from vocabulary tests for their typical meanings. In *Proc. of Educational Data Mining (short paper)*, 2022.

[4] M. Martinc, S. Pollak, and M. Robnik-Šikonja. Supervised and Unsupervised Neural Approaches to Text Readability. *Computational Linguistics*, 47(1):141–179, Apr. 2021.

[5] L. Pan, X. Wang, C. Li, J. Li, and J. Tang. Course concept extraction in MOOCs via embedding-based graph propagation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 875–884, Taipei, Taiwan, Nov. 2017. Asian Federation of Natural Language Processing.

[6] S. Vajjala and I. Lučić. OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 297–304, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[7] J. Yu, G. Luo, T. Xiao, Q. Zhong, Y. Wang, W. Feng, J. Luo, C. Wang, L. Hou, J. Li, Z. Liu, and J. Tang. MOOCCube: A Large-scale Data Repository for NLP Applications in MOOCs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, July 2020.

[8] J. Yu, C. Wang, G. Luo, L. Hou, J. Li, J. Tang, M. Huang, and Z. Liu. ExpanRL: Hierarchical Reinforcement Learning for Course Concept Expansion in MOOCs. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 770–780, Suzhou, China, Feb. 2020. Association for Computational Linguistics.

[9] H. Zhang, X. Shen, B. Yi, W. Wang, and Y. Feng. KGAN: Knowledge Grouping Aggregation Network for course recommendation in MOOCs. *Expert Systems with Applications*, 211:118344, Jan. 2023.

Pre-selecting Text Snippets to provide formative Feedback in Online Learning

Sylvio Rüdian
Humboldt-Universität zu Berlin
ruediasy@informatik.hu-berlin.de

Clara Schumacher
Humboldt-Universität zu Berlin
clara.schumacher@hu-berlin.de

Jakub Kuzilek
Humboldt-Universität zu Berlin
jakub.kuzilek@hu-berlin.de

Niels Pinkwart
German Research Center for Artificial Intelligence
niels.pinkwart@dfki.de

ABSTRACT

In this paper, a proof of concept is shown to generate formative textual feedback in an online course. The concept is designed to be suitable for teachers with low technical skill levels. As state-of-the-art technology still does not provide high-quality results, the teacher is always held in the loop as the domain expert who is supported by a tool, and not replaced. The paper presents results of our proposed approach for semi-automatic feedback generation using a real-world university seminar, where students create sample micro-learning units as online courses, for which they get feedback for. A supervised machine learning approach is trained based on learner submissions features, and the feedback, that was chosen by teachers in former submissions. The results are promising.

Keywords

Formative Feedback, Online Learning, Teacher Support, Prediction.

1. INTRODUCTION

Feedback is considered essential for supporting successful learning processes and outcomes [1]. Feedback can be defined as „information provided by an agent (e.g., teacher, peer, book, parent, self, experience) regarding aspects of one's performance or understanding" [1]. However, the timely provision of elaborated individual feedback is limited due to large student cohorts and limited resources in higher education. The lack of resources results in predominant use of summative assessments (and feedback) [2], which are often used at the end of a learning unit or course for grading and certification purposes if predefined objectives are met [3]. Due to heterogeneous students, the provision of individual support is even more relevant. Therefore, formative assessments aiming at providing students feedback on their performance or next learning steps is crucial. Instead of being distinct concepts, the functions of formative and summative assessments are on a continuum as such that the engagement with assessment tasks or the potential feedback can result in a change of learners' behavior [4]. In sum, elaborated feedback offering information on task-, process- and self-regulation level has been found to be most effective for learning success [5]. This includes an understanding of the learning goals that need to be achieved („Where the learner is going"), assessing the evidence of learning („Where the learner is right now"), and the provision of feedback on how to achieve the designated learning goals [6]. However, even with the increased use of digital learning environments and methods such as learning analytics the provision of informative feedback at scale is challenging [7] and time-consuming. Due to the need of extra resources, formative feedback is often not provided at all, or solely on the correctness, in the form of sample solutions or short paragraphs. The paper aims to support teachers in the process of giving textual feedback.

2. RELATED WORK

Automated feedback can be characterized based on several properties [8]: a) the adaptiveness of the feedback; b) its timing; c) learners' control over the feedback: and d) the purpose of the feedback. Automated feedback can for example be not adaptive at all, dependent on students' solution to a task or also on their characteristics and learning behavior. Timing of the automated feedback can be immediate after the action, upon request or at the end of a task. The feedback provision might further be controlled by the learner for example with regards to the amount and frequency of feedback, the timing, its appearance. The need for control has also been brought up by studies investigating students' preferences of automated interventions (e.g. [9]). The purpose of the feedback refers to simple corrective feedback, suggestion of future actions, additional information, or motivational feedback [8]. Despite the examination of computer-generated feedback for decades; still the creation of highly informative feedback is very complex, where machines can be supportive, but do not replace teachers [10]. As texts created by learners are manifold and diverse it is hard to evaluate them automatically [11]. Due to the low quality of computer-generated feedback, its use can lead to high frustration [12]. For example, available state-of-the-art automatic writing evaluation tools, such as proofreading tools to detect mistakes in submissions of language learners, do not meet teachers' expectations [13]. Hence, the teacher is vital for the provision of feedback. Thus, instead of providing computer-generated feedback to learners directly, a teacher-in-the-loop approach is of high importance. Therefore, the process to create feedback must be intuitive without the need for complex adjustments.

In the domain of education, decisions coming from computer-generated feedback tools must be explainable. This is a key component of the trusted learning analytics approach (TLA) [14]. One possible solution is the tool OnTask [15], which can principally be used to generate texts based on pre-defined text snippets and rules that use trace data. Based on such rules, decisions can be justified and explained. If for example, the learner submits a text and the tool recognizes that the learner skipped watching a related learning video, which is implemented as a rule, then feedback is given using the snippet with the advice to have a deeper look at the learning

S. Rüdian, C. Schumacher, J. Kužílek, and N. Pinkwart. Pre-selecting text snippets to provide formative feedback in online learning. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 430–433, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115748

material. However, it is essential to educate teachers so that they get an understanding of the versatility of such software. Teachers must have scenarios in mind, which must be implemented in rules. From the practical perspective, this is a pitfall as teachers want to focus on their domain to create learning material and not on scenarios that possibly can exist [12]. Hence, feedback is mainly limited to tasks, where feedback can be predefined. For multiple-choice questions, feedback can be given if the correct choices are selected, but also for incorrect selections, respectively. Considering textual submissions, the state-of-the-art Moodle, and H5P versions allow searching for specific keywords. If they are missing in the text, feedback can be provided. Nevertheless, such feedback assumes that the learner uses concrete vocabulary (or synonyms, that are predefined by the teacher). If they use other words or descriptions, they still get the same feedback as others, which can lead to frustration. If learner texts are aimed to be evaluated on an individual level automatically, the topic of automatic essay scoring (AES) emerges. There, texts are scored, intending to compare learners' results. Most AES systems have in common, that they need to be trained with a large sample size with annotated texts and they extract a huge number of linguistic features [16]. Exemplarily, the AES „IntelliMetric" [17] extracts over 300 features, ranging from conceptual, and structural features to rhetorical attributes [18]. First, the approach examines cohesiveness and consistency. Then, the scope of the content is analyzed, followed by an evaluation of text structure, and transitional fluency. Then, sentence structure is investigated, using sentence complexity with readability metrics, and syntactic variety. Finally, mechanics and conventions are analyzed, to test whether the text is in line with standard American English (spelling, grammar, etc.) [16]. However, most tools are not open-source and rely on financial benefits. Thus, their application is limited to institutes, which have the budget to spend.

3. FRAMEWORK

Following Deeva et al. [8], automated feedback can be expert-driven as in the rule-based systems (e.g., OnTask), or data-driven considering student data using algorithmic approaches or a combination of both. In the proposed framework, the importance of the teacher in the loop is emphasized. The idea of having the teacher-in-the-loop is extended by Rüdian et al. [19] to connect learner submissions with feedback by exploring derived NLP features and its relation to feedback given in concrete contexts. To the best of our knowledge, this concept has not been applied to a real-world online course setting. Thus, we focus on the research question of whether there is a set of NLP features (extracted from learner submissions), that are predictive to auto-select ratings, which were previously selected by teachers.

The approach proposes a teacher-in-the-loop approach that is based on pre-defined text snippets to provide feedback on task-level. Such snippets can be extracted from already given feedback texts or best practices in the literature. Text snippets must meet the condition to be related to a scale (e. g., Likert scale, binary (yes/no) scale). In the training process, teachers create feedback by selecting pre-defined text snippets. The idea of using such snippets is not new, but a helpful step for teachers to reduce the required time to create feedback [19]. Then, snippets are stored including the rating on the scale, e. g. whether a learner correctly applied a concept, or not. NLP features are extracted from user artifacts (e. g. textual submissions). Features can be based on sentiment analysis, word-sense disambiguation, argument mining, or others [18]. Such features are then used to train a supervised machine learning approach, aiming to predict ratings on evaluation criteria. Explainable methods such as the Naïve Bayes classifier [20] are favored to follow the TLA approach. For all labels that can be predicted with acceptable accuracy, a model is stored. Then, for new learner artifacts of the same task, ratings can be predicted. The teacher gets those predictions so that related text snippets are automatically pre-selected when the teacher aims to create feedback. Based on those selections, a final feedback text is generated. Besides, a reinforcement learning approach is used. The teacher can change pre-selections. Thus, new training data are continuously created to train the model with more data to become more generalizable. Also, the student can evaluate feedback to obtain a critical view of its applicability. The main idea is to separate teachers from the machine learning approach, that runs in the background.

4. STUDY DESIGN

In a university seminar, students have the task to design a micro-learning online course (~15-25 min) covering a topic of their choice. Students create courses in a Moodle instance. 33 courses are created. They receive feedback from a tutor who uses a form of 28 evaluation criteria and selects whether the criteria are fulfilled. Selections must be rated on a Likert (5=totally agree to 1=totally disagree) or binary scale (the latter is used for the case, where only two options exist). For binary options, also 5 (agree), and 1 (do not agree) are used. Feedback criteria are based on literature research to rate the quality of an online course. In detail, clearness, instructions, and learning materials are rated, whether appropriate feedback is given [23], learning goals and expectations are included [24], a target group is defined, and whether the course content is appropriate for those learners [25]. Further, it is rated whether designed tasks have an appropriate difficulty level and whether final tests are suitable to evaluate knowledge gain [26], and, of course, the correctness of the created learning material.

The tutor uses the system to generate a feedback text, based on his/her selections, which is the standard process in this setting to provide feedback. The automatically generated text can be changed or enhanced by the tutor. However, as to date, further text adjustments are only used to a negligible amount by the tutors; this will be investigated at a later stage in more detail and is not covered in this paper. Selected feedback options are stored for each course that is submitted by students. Those courses are the artifacts and build the base for the data set. Thus, the courses are used as the input variables and the aim is to pre-select the rating on the evaluation criteria, that are used to generate the textual feedback.

Then, an experimental analysis is done to examine the predictability of the items. Textual features must first be extracted from all courses. To do that, courses are transferred to a CSV file using Moodle backups of the courses, and from that, the main information is extracted. Each line is related to an item of the course progression. The CSV file contains the item type (more detailed, whether H5P is used, a content page is created, or the Moodle quiz tool is used). It contains the header of the item, the content, and in case of interactive items (H5P/quiz), also questions including responses, correctness, and feedback. Based on that information, the course can principally be reconstructed. As a CSV file is created for each course, a transfer to a feature vector is required, containing the same number of features for each course, aiming to train a predictive model.

The following features are extracted and stored in a new CSV file:

(1) Number of items, including types (H5P, pages, Moodle quiz, videos),
(2) Text complexity metrics of the content, and questions (Flesh Reading Ease [25], or Gunning Fog Index [26]),

(3) Use of keywords in texts („target group", „references/literature"),
(4) Number of items, where feedback is given, namely feedback given on wrong, or correct responses, and overall feedback,
(5) Polarity and subjectivity of contents.

Before training an approach to make predictions, the distribution of selected options is analyzed to detect highly imbalanced options. Due to the limited number of courses (33), ratings on a Likert scale are not well balanced. To be still able to predict those ratings, ratings are transformed to a binary scale on indicating that the criterion is met (5, criterion passed) and one representing all other values (1-4; at-risk, criterion failed). Criteria, that are still imbalanced (like all students fulfilled them), are filtered out as it is not worth examining predictability due to the limited dataset. To give an example: All students described the learning goal in their course. Thus, there is no low rating for the criterion, so it can be ignored. For the remaining ones, distributions are explored to see whether there is a remarkable difference, considering all features separately. Those, where a difference can be seen, are selected for the proof-of-concept. Then, a Naïve Bayes model is trained, as it is easy to interpret probabilities, which fulfills the TLA condition. Besides, it can easily be extended to a multidimensional problem and the resulting trained model can easily be implemented by using web technologies without the necessity of deploying complex computational power. The resulting predictions are evaluated using 5-fold cross-validation.

5. RESULTS

Based on the initial analysis of the distributions of binary ratings for all criteria and all features, the target variable is chosen on whether *the learning goal is covered by a final test*. There have been 11 failed and 22 passed cases. For the selected target variable, the corresponding extracted features distributions with a focus on the target class has been visually analyzed and the most promising 4 features have been selected for the initial proof-of-concept. Selected features are: the number of Moodle quiz items; the number of feedbacks given for correct responses; question readability grades ARI (Automated Readability Index); and question readability grades for Flesch Reading Ease (FRE). Remaining features are excluded. The corresponding distributions are depicted in Figure 1.

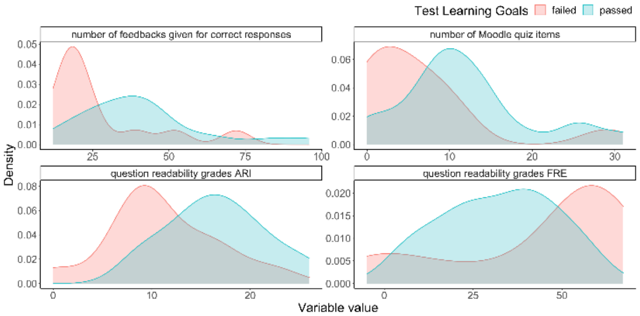


Figure 1: Distributions of binary classes for four features.

Both cases (passed vs. at-risk/failure) are plotted with red, and blue colors. For selected features, differences in distributions can be seen. This is a good sign, as those features split the dataset by the binary ratings in general. Following the selected features, the Naïve Bayes model with Gaussian kernel is trained using 10-fold cross-validation. The error is estimated using 5-fold cross-validation covering the complete data set. Thus, the process of error estimation and model training is as following: data is divided into 5 folds using stratified sampling without replacement and in 5 steps, the model is trained using 4 folds of input data via 10-fold cross-validation and the error is estimated with the remaining 1 data fold.

To simplify the model for the deployment, we explored 5 different scenarios: using each feature separately (4 scenarios) and using selected features together to train the model. Table 1 reports the results using mean values accuracy (Acc), precision (P), and recall (R) in 5 rounds of cross-validation. P and R are computed for both classes (passed and at-risk/failed) to understand their predictive power (guessing would be .5 for the binary option). As visible, the feature of the „number of given feedback on correct responses" outperforms scenario 5, where all features are used together.

Table 1. Results for four features.

Feature	Acc	P	R	P	R
		passed		failed	
Number items quiz	.68	.74	.78	.63	.46
Number of given feedback on correct responses	.75	.87	.77	.63	.73
Question readability ARI	.76	.80	.86	.81	.56
Question readability FRE	.73	.75	.90	.60	.43
All features	.71	.77	.77	.63	.60

6. DISCUSSION

Compared to a pre-defined rule-based approach the proposed approach allows to provide more fine-grained feedback and dynamic support. Furthermore, it aims at enhancing teachers' practices and reducing their workload for providing highly informative feedback on text artifacts. Thus, the approach considers the limited resources in higher education for providing formative feedback but still enables learners to derive appropriate future learning activities. Due to complexity of algorithms and their limitedness of providing actionable outcomes a major concern in educational settings is the limited acceptance of the stakeholders. This might be avoided by the simplicity of the proposed approach that enables teachers to create feedback without the need for abstract technical skills plus by being grounded in the idea of TLA of having the human in the loop of an explainable approach.

This proof-of-concept is limited as only data of students that agreed to share their data for this research were analyzed resulting in 33 submissions which might have led to biases. This calls for future research with larger data sets.

From a statistical perspective, computational complexity involves the estimation of the Gaussian distribution during the model training and then, it compares two posterior probabilities. In the final model, only two equations (for estimation of the probabilities) and their comparisons are computed. We also limited ourselves to the most promising features and selected one criterion for which the concept is working well. The training step is required for each criterion, requiring to create 28 separate models. Thus, in future a more refined approach which can do the estimation at once (for example by mapping the separate criteria to another dimension, where one number reflects unique criteria combination) will be explored. The restriction to binary cases is necessary to simplify the small dataset, in future work either the one-vs.-rest/one-vs.-one approach or a regression model for proper estimation of the scale values will be investigated. However, if this is aimed to be examined, the dataset must be extended with further samples.

The accuracy of using all features suggests, that the model tends to overfit with higher dimensions. Using one feature leads to better results, less computational complexity, and ease of use. Values of precision and recall are better in general for the case of passed class. This is probably because even with the selected binary target value, classes are imbalanced. Thus, the trained model prefers positive cases due to a better fit of the distribution. Further, we used a handful of NLP features, extracted from learner submissions. Exploring more linguistic features is of high interest to explore its predictive power. However, the approach needs to be enhanced further to support the teacher more as still the human needs to validate the feedback which might be also time-consuming. Furthermore, also students' behavioral data should be considered, for example, to determine the timing of the feedback as well as its properties (e.g., provided by a system or an e-mail of the tutor (see further [27]). As the uptake and actual use of feedback by the students is key for its effectiveness [28], their perceptions of the feedback [7] as well as their actions taken need to be investigated in more detail. Using experimental study designs the impact of the feedback on students' learning processes and outcomes will be investigated in detail. In sum, the proof-of-concept is promising, and predictions have been working in the concrete setting.

7. ACKNOWLEDGEMENTS

This work was supported by the German Federal Ministry of Education and Research (BMBF), grant number 16DHBKI045.

8. REFERENCES

[1] J. Hattie and H. Timperley, "The power of feedback" in *Review of educational research 77(1)*, 2007, pp. 81-112.

[2] J. Broadbent, E. Panadero and D. Boud, "Implementing summative assessment with a formative flavour: a case study in a large class" in *Assessment and Evaluation in Higher Education 43(2)*, 2017, pp. 307-322.

[3] V. J. Shute and B. J. Becker, "Prelude: Assessment for the 21st century" in *V. J. Shute & B. J. Becker (Eds.), Innovative Assessment for the 21st Century. Supporting Educational Needs* , Springer, 2010, pp. 1-11.

[4] P. Black and D. Wiliam, "Classroom assessment and pedagogy" in *Assessment in Education: Principles, Policy & Practice, 25(6)*, 2018, pp. 551-575.

[5] B. Wisniewski, K. Zierer and J. Hattie, "The power of feedback revisited: A meta-analysis of educational feedback research" in *Frontiers in Psychology 10*, 2020, p. Article: 3087.

[6] D. Wiliam and M. Thompson, "Integrating assessment with learning: What will it take to make it work" in *C. A. Dwyer (Ed.), The Future of Assessment. Shaping Teaching and Learning*, Lawrence Erlbaum Associates, 2008.

[7] L.-A. Lim, S. Dawson, D. Gašević, S. Joksimović, A. Fudge, A. Pardo and S. Gentili, "Students' sense-making of personalized feedback based on learning analytics" in *Australasian Journal of Educational Technology 36(6)*, 2020, pp. 15-33.

[8] G. Deeva, D. Bogdanova, E. Serral, M. Snoeck and J. De Weerdt, "A review of automated feedback systems for learners: Classification framework, challenges and opportunities" in *Computers & Education (162), 104094.*, 2021.

[9] C. Schumacher and D. Ifenthaler, "Features students really expect from learning analytics" in *Computers in Human Behavior (78)*, 2018, pp. 397-407.

[10] E. C.-F. Chen and W.-Y. E. C. Cheng, "Beyond the design of automated writing evaluation: Pedagogical practices and perceived learning effectiveness in EFL writing classes" in *Language Learning & Technology 12.2*, 2008, pp. 94-112.

[11] T. K. Landauer, "Automated scoring and annotation of essays with the Intelligent Essay Assessor" in *AES: A cross-disciplinary perspective*, 2003, p. 87–113.

[12] P. Ware, "Computer-generated feedback on student writing" in *Tesol Quarterly 45.4*, 2011, pp. 769-774.

[13] S. Rüdian, M. Dittmeyer and N. Pinkwart, "Challenges of using auto-correction tools for language learning" in *LAK22*, 2022, pp. 426-431.

[14] J. Hansen, C. Rensing, O. Herrmann and H. Drachsler, "Verhaltenskodex für Trusted Learning Analytics. Version 1.0. Entwurf für die hessischen Hochschulen" in *Innovationsforum Trusted Learning Analytics*, 2020.

[15] A. Pardo, K. Bartimote, S. B. Shum, S. Dawson, J. Gao, D. Gašević and L. Vigentini, "OnTask: Delivering data-informed, personalized learning support actions" 2018, pp. 235-249.

[16] S. Dikli, "Automated Essay Scoring" in *Turkish Online Journal of Distance Education-TOJDE (7)*, 2006.

[17] L. M. Rudner, V. Garcia and C. Welch, "An evaluation of IntelliMetric™ essay scoring system" in *The Journal of Technology, Learning and Assessment 4.4*, 2006.

[18] intellimetric, "How it Works" Vantage Labs, 17 06 2017. [Online]. Available: https://www.intellimetric.com/how-it-works. [Accessed 05 01 2023].

[19] S. Rüdian. C. Schumacher, N. Pinkwart: "Computer-Generated formative Feedback using pre-selected Text Snippets" in *LAK*, 2023, pp. 129-131.

[20] K. P. Murphy, "Machine learning: a probabilistic perspective", MIT press, 2012.

[21] Q. Matters, "K-12 Rubric Workbook Standards for Course Design (Fifth Edition)", Annapolis, MD: Maryland Online, 2019.

[22] D. Xu, Q. Li and X. Zhou, "Online course quality rubric: a tool box" in *Online Learning Research Center*, 2020.

[23] S. Baldwin, Y. H. Ching and Y. C. Hsu, "Online course design in higher education: A review of national and statewide evaluation instruments" in *TechTrends 62(1)*, 2018, pp. 46-57.

[24] S. D. Achtemeier, L. V. Morris and C. L. Finnegan, "Considerations for developing evaluations of online courses" in *Journal of Asynchronous Learning Networks 7.1*, University of Georgia, 2003, pp. 1-13.

[25] R. Flesch, "A new readability yardstick" in *Journal of applied psychology, 32*(3), 221, 1948.

[26] R. Gunning, "The Technique of Clear Writing", McGraw-Hill, 1952.

[27] C. Schumacher, D. Ifenthaler, "Investigating students' perceptions of system- vs. teacher-based learning analytics feedback", 2023.

[28] N. E. Winstone, R. A. Nash, M. Parker and J. Rowntree, "Supporting learners' agentic engagement with feed-back: A systematic review and a taxonomy of reciepience processes" in *Educational Psychologist 52(1)*, 2017, pp. 17-37.

Exploring the Implementation of NLP Topic Modeling for Understanding the Dynamics of Informal Learning in an AI Painting Community

Ran Bi*, Shiyao Wei*

SAS Institute, Florida State University

ran.bi@sas.com,

sw22b@fsu.edu

ABSTRACT

Informal learning is a significant part of lifelong learning. The rise of online communities as a new venue for informal learning has led to an increase in the availability of discourse data. As the dataset grows, it is feasible for scholars to understand the learning dynamics of these communities. However, the manual coding and analysis of such large datasets can be cost-prohibitive. Natural Language Processing (NLP) has been demonstrated to be a viable solution for analyzing large datasets in educational contexts. In this paper, we explore the application of NLP topic modeling method, Latent Dirichlet allocation (LDA), in understanding informal learning dynamic within an AI painting community. We collected data in two months from November 7, 2022, to January 8, 2023, and our findings show that major topics discussed in the space are around ethics, models, and procedures of AI painting, and topics updated over two months.

Keywords

Topic modeling, Affinity Space, LDA, AI Painting, Informal Learning

1. INTRODUCTION

The first and second decades of the 21st century have seen the emergence of online communities, such as subreddits on Reddit and groups on Facebook. These communities provide a platform for interest-driven learning outside of formal education settings [11]. Studies have shown that online spaces, particularly those in remote areas, provide individuals with a shared space to learn diverse knowledge [4], such as literacy [3] and disease management [15]. Additionally, online affinity spaces bridge the gap between socioeconomic, ethnic, and social groups, allowing learners to communicate freely around topics of interest [4].

As the Internet becomes more prevalent, there is a growing amount of data available for studying online affinity spaces [8]. However, as the size of data increases, so does the cost of hand-coding it. Traditional qualitative coding requires researchers to read and understand thousands of data points [12], which can be costly and time-consuming. To address this issue, researchers have been exploring alternative methods, such as natural language processing (NLP), to get a snapshot of the data before embarking on the hand-coding process [10].

NLP has been shown to be a valid solution for qualitative research. Previous research [9] has demonstrated that when used with appropriate parameters, NLP can effectively enhance our ability to systematically investigate and interpret discourses in large collections of text.

AI painting is a new application of generative AI. It works by using algorithms to analyze and learn from images available on the Internet and input specified by humans [13]. The algorithm generates new images in adherence to the aesthetics it has learned. AI painting has attracted public attention and sparked many discussions in online communities due to its potential and associated risks. For example, the subreddit r/StableDiffusion is a prevalent community where participants gather, share, ask, and debate around AI painting issues. In previous work, we hand-coded 2,291 posts and comments in r/StableDiffusion and found eight major topics of discussion: algorithm & model, application, data, entertaining, ethics & social implications, hardware, off-topic, and procedure. In this paper, we use Latent Dirichlet allocation to identify the major topics discussed in the space and determine if there are changes within 2 months.

2. RELATED WORK

2.1 Affinity Space and Informal Learning

Online affinity spaces have garnered the attention of researchers in education field. Affinity spaces are a form of public pedagogy in informal learning [6]. In these spaces, learners exchange information about shared passions through design and resources [6, 15]. Affinity spaces help learners prepare for their lifelong learning journey outside of traditional educational environments.

Discussions play a crucial role in the information-exchange process within affinity spaces. Understanding the dynamics of these discussions is essential for studying informal learning. Recent studies on the discussion patterns of affinity spaces [14, 15] have identified key content types on online social network sites and different behaviors between key and other actors. While [14] collected 514 posts discussing disease management, they did not examine the change of topics over time. Additionally, previous

R. Bi and S. Wei. Exploring the implementation of nlp topic modeling for understanding the dynamics of informal learning in an ai painting community. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 434–437, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115753

studies have not addressed the issue of how the topics in a technology-focused affinity space change. Through this study, we aim to utilize topic modeling to analyze the larger scale of discourse data and identify the dynamics of the space.

2.2 Topic Modeling in Discourse Analysis

Topic modeling is an NLP method applied in discourse analysis. One of the most widely used topic modeling methods is Latent Dirichlet Allocation (LDA), which derives probabilities of words belonging to topics (clusters of semantically related words) from textual data [21]. Another study [20] investigated the potential of using LDA to explore topics emerged in social media data during the COVID-19 pandemic and found that LDA is useful for stakeholders to understand the most discussed topics in the field.

The gap in literature identified frames our research question and the methods discussed above contribute to the choice of our research methods. Our research questions are as follows:

RQ1: *What are the topics most discussed in the subreddit in 2 months?*

RQ2: *How do topics change in an AI painting affinity space in 2 months?*

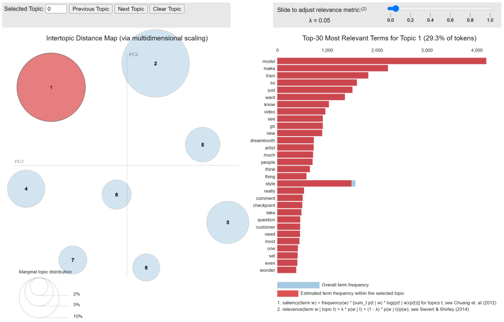
3. METHOD

3.1 Data Collection and Cleaning

In this research, we aim to observe the topics discussed and how topic changed in an AI painting affinity space. Thus, we chose one highly frequented platform for discussion surrounding AI-generated painting as the observation site. We observed the community for 2 months, utilizing the Pushshift API [1]. All posts and comments were obtained within a specific time frame, from November 7th, 2022, to January 8th, 2023. The sample included 14,319 posts and 172,770 comments. The 2-month period included workdays, weekends, and vacation season to help us better understand the trend and dynamics of informal learning in the space. In terms of data cleaning, a flexible approach was implemented. Firstly, comments were removed if they were not associated with posts within the designated time frame. Secondly, all posts were concatenated based on their title, text, and comments, being treated as a single paragraph.

3.2 Data Analyzing and Visualization

In the process of analyzing data, a time series analysis was conducted on the number of posts, comments, and subscribers of the Stable Diffusion channel over a period of two months (9 weeks). It indicates an increase in subscribers from 80,000 to 116,000 at a steady rate. Furthermore, the time series plot of comments revealed three peaks during the two-month period. The first peak occurred during the week of Thanksgiving, due to the viral spread of AI-generated holiday greeting graphs. The other two peaks occurred two weeks prior to Christmas. Based on our analysis, we noticed a seasonality of increased comments on weekends as opposed to a higher frequency of post submissions on weekdays. Additionally, text mining and topic modeling using LDA was conducted, yielding interesting results that warrant further investigation. Data visualization is achieved by using LDAvis [16], an interactive visualization of topics estimated using LDA. As shown in Figure 1, bubble graph refers to different topics emerged from the material, with a red bubble highlighted. The bar chart refers to the frequency of top 30 terms related to topic 1 that appeared in the context of topic 1. The slide bar could adjust the relevance parameter of terms. The numbers of week mentioned in this paper represent the order of week in a year, for example, week 45 is the 45th week in 2022.



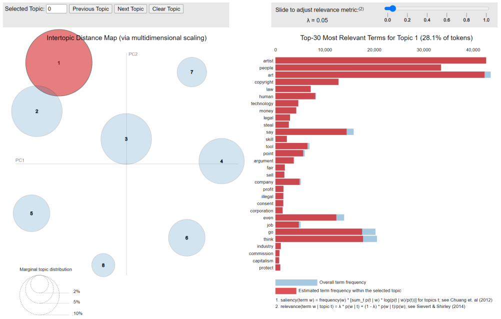
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 1. Topic modeling of posts-only data

4. RESULTS

4.1 What are the Topics Most Discussed in the Subreddit in 2 Months?

When we examined the results of posts only, topics related to models, such as training the model and Dreambooth, are the most discussed. Ethical and social implications, including keywords such as art and artists, are also mentioned in the first category. However, when analyzing both posts and comments, the topics become clearer, as shown in Figures 1 and 2.



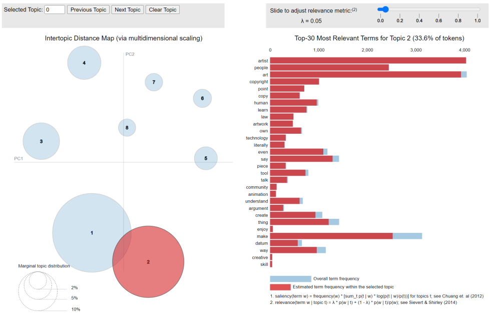
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 2. Topic modeling of posts and comments data

In the results of posts-and-comments, topics related to ethics, such as artists and art, are separated from the previous first category. We found that the results of LDA partly align with the hand-coding results from our previous research. In the results of all posts and comments, the second most discussed topic is about models, which contains words such as "model", "train", and "training". In topic 3, words related to procedures, such as "run", "file", "folder", and "download" cluster together. In topic 5, words related to applications, such as "video", "game", and "life" emerged. Topic 6 is about all prompts used in the process of generating images, such as "prompt", "picture", "text", "girl", and "man".

4.2 How did Topics Change in an AI painting Affinity Space in 2 Months?

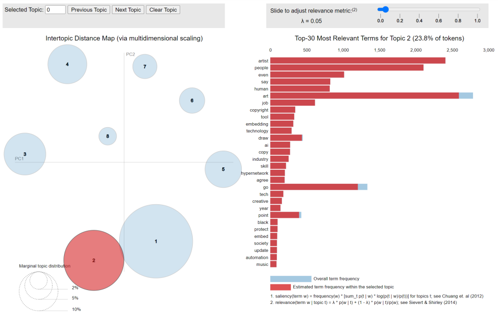
In our observation of weekly differences, we found that in Week 45, the discussion of ethics in Topic 2 focused on the issue of copyright, as shown in Figure 3. By adjusting the relevance metric to 0, we observed an increase in the weight of the keyword "copyright".



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 3. Topic modeling result from week 45

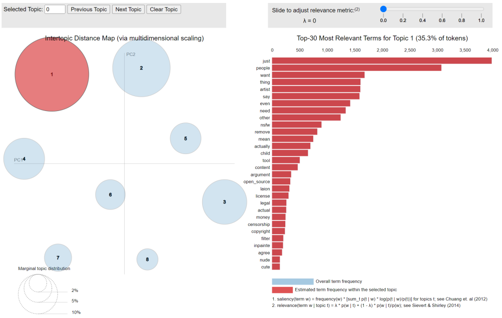
In Week 46, as shown in Figure 4, participants were more concerned about job and industry in the context of ethics. Additionally, compared to the previous week, there was a new discussion on watermarks, due to the release of Stable Diffusion 2 (SD2) which tends to generate images with watermarks, which many users were complaining about.



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 4. Topic modeling result from week 46

In Week 47, as shown in Figure 5, we noticed a different keyword, "censorship." Upon further examination of the original data, we found that it was related to the release of SD2.



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 5. Topic modeling result from week 47

5. DISCUSSIONS

In alignment with previous research [5, 18], in this paper, we believe that LDA can only provide a glimpse of the data and capture certain keywords in the discussion. To gain more in-depth insights, researchers must go back to the data and explore the reasons behind why these keywords appear. However, LDA can save time by reducing the need to read less important data and improve the efficiency of analyzing discourse data in social media. Affinity space as an information hub, dispersed knowledge pattern, time-sensitivity of topics, and limitations and future research are discussed below.

5.1 Information Hub for Interested Learners

Affinity spaces serve as an information hub for all interested learners, and our results show that they contain mainstream topics related to AI painting, aligning with our previous hand-coding results. Topics related to ethics and social implications, such as art, artists, industry, jobs, and copyright, consistently took the first or second position throughout 2-month data. Similarly, topics related to applications consistently appeared throughout 2 months, although with a lower ranking. Procedures and algorithm & models were the second most discussed topics. Also, users participated in much discussion about prompts and data in the subreddit, echoing the open-source tradition in coding community [7]. Discussions about hardware appeared in several weeks and some keywords related to hardware were mixed in the algorithm and models category. Some entertaining content was also mixed in off-task categories.

According to [4], the subreddit r/StableDiffusion acts as an affinity space where learners can share their experiences and knowledge surrounding AI-generative painting, specifically about Stable Diffusion. This generator allows individuals to gather and explore sets of signs and potential relationships among signs [4]. Unlike other learning environments such as bootcamps that cater to a specific level of skill, r/StableDiffusion welcomes both beginners and experts alike [4]. The community encourages both extensive and intensive learning [4], with members frequently sharing analyses of results, algorithms, and models, providing abundant resources for novice learners entering the space.

5.2 Dispersed Knowledge of Affinity Space

During our topic modeling analysis, we observed that learners engage in sharing behaviors that connect to other sites, such as "png" and "github", which are external to the subreddit. This reveals that space enables users to actively participate in sharing and learning beyond its confines. The distributed nature of knowledge in network-like formats means that there are no strict boundaries or limitations on what learners can access, which encourages their agency [19]. Freedom and choice are vital components of informal learning [17], and the dispersed nature of the resources connected by a network-like format enables learners to select what interests them the most, and continue their learning journey accordingly.

5.3 Time-sensitiveness of Affinity Space

We found that the affinity space is time-sensitive, meaning that users promptly respond to updates of Stable Diffusion and other related news in AI painting. For example, even before the public release of Stable Diffusion 2 on November 24, 2022, users were discussing the watermark issue in SD2 in Week 46 (November 7-13, 2022). Similarly, the launch of ChatGPT on November 30, 2022 was also discussed in the following week's discussion. This finding adds to current understanding of affinity space and informal learning, especially a few principles mentioned in [4]. Learners in affinity space proactively react to the development and updates of the software and might change the development of the software itself.

Time-sensitiveness matters because it contributes to the learning material development in the affinity space. Affinity space is a public pedagogy [6]. People come here for up-to-date experience and knowledge, thus, the immediacy of sharing and response to the software in the space are useful learning materials for all learners in the space.

5.4 Limitations and Further Research

While our analysis captured the topics that emerged during the two-month period of our data collection, we acknowledge that the dynamics of the community are constantly evolving. In addition, comparing the keywords from each week proved challenging due to the sheer volume of data. Future research could benefit from using dynamic topic modeling [2], another NLP methods in discourse analysis, to achieve a more in-depth understanding of how the community's discourse and topics of discussion evolve over time.

6. CONCLUSION

In this paper, we report on the progress of using Latent Dirichlet Allocation (LDA) to capture the dynamics of topics in an AI painting affinity space. We collected data over a two-month period, from November 7, 2022, to January 8, 2023. Our findings indicate that the community's primary topics revolve around ethics, models, and procedures, and that these topics evolved over the course of the two-month period.

7. REFERENCES

[1] Baumgartner, J. et al. 2020. The Pushshift Reddit Dataset. *Proceedings of the International AAAI Conference on Web and Social Media*. 14, (May 2020), 830–839. DOI:https://doi.org/10.1609/icwsm.v14i1.7347.

[2] Blei, D.M. and Lafferty, J.D. 2006. Dynamic topic models. *Proceedings of the 23rd international conference on Machine learning - ICML '06* (New York, New York, USA, 2006), 113–120.

[3] C. Lammers, J. et al. 2012. Toward an affinity space methodology: Considerations for literacy research. *English Teaching*. 11, 2 (Jul. 2012), 44-n/a.

[4] Gee, J.P. 2005. Semiotic social spaces and affinity spaces: from *The Age of Mythology* to today's schools. *Beyond Communities of Practice*. Cambridge University Press. 214–232.

[5] Gencoglu, B. et al. 2023. Machine and expert judgments of student perceptions of teaching behavior in secondary education: Added value of topic modeling with big data. *Computers & Education*. 193, (Feb. 2023), 104682. DOI:https://doi.org/10.1016/j.compedu.2022.104682.

[6] Hayes, E.R. and Gee, J.P. 2010. *Handbook of Public Pedagogy*. Routledge.

[7] Heller, B. et al. 2011. Visualizing collaboration and influence in the open-source software community. *Proceedings of the 8th Working Conference on Mining Software Repositories* (New York, NY, USA, May 2011), 223–226.

[8] Hewson, C. 2020. Qualitative Approaches in Internet-Mediated Research: Opportunities, Issues, Possibilities. *The Oxford Handbook of Qualitative Research*. Oxford University Press. 633–673.

[9] Jacobs, T. and Tschötschel, R. 2019. Topic models meet discourse analysis: a quantitative tool for a qualitative approach. *International Journal of Social Research Methodology*. 22, 5 (Sep. 2019), 469–485. DOI:https://doi.org/10.1080/13645579.2019.1576317.

[10] Nelson, L.K. et al. 2021. The Future of Coding: A Comparison of Hand-Coding and Three Types of Computer-Assisted Text Analysis Methods. *Sociological Methods & Research*. 50, 1 (Feb. 2021), 202–237. DOI:https://doi.org/10.1177/0049124118769114.

[11] Peters, M. and Romero, M. 2019. Lifelong learning ecologies in online higher education: Students' engagement in the continuum between formal and informal learning. *British Journal of Educational Technology*. 50, 4 (Jul. 2019), 1729–1743. DOI:https://doi.org/10.1111/bjet.12803.

[12] Prior, L. 2020. Content Analysis. *The Oxford Handbook of Qualitative Research*. P. Leavy, ed. Oxford University Press.

[13] Reed, S. et al. 2016. Generative adversarial text to image synthesis. (2016), 1060–1069.

[14] Sharma, P. et al. 2021. Knowledge sharing discourse types used by key actors in online affinity spaces. *Information and Learning Sciences*. 122, 9/10 (Sep. 2021), 671–687. DOI:https://doi.org/10.1108/ILS-09-2020-0211.

[15] Sharma, P. and Land, S. 2019. Patterns of knowledge sharing in an online affinity space for diabetes. *Educational Technology Research and Development*. 67, 2 (Apr. 2019), 247–275. DOI:https://doi.org/10.1007/s11423-018-9609-7.

[16] Sievert, C. and Shirley, K. 2014. LDAvis: A method for visualizing and interpreting topics. *Proceedings of the workshop on interactive language learning, visualization, and interfaces* (2014), 63–70.

[17] Song, D. and Bonk, C.J. 2016. Motivational factors in self-directed informal learning from online learning resources. *Cogent Education*. 3, 1 (Dec. 2016), 1205838. DOI:https://doi.org/10.1080/2331186X.2016.1205838.

[18] Törnberg, A. and Törnberg, P. 2016. Muslims in social media discourse: Combining topic modeling and critical discourse analysis. *Discourse, Context & Media*. 13, (Sep. 2016), 132–142. DOI:https://doi.org/10.1016/j.dcm.2016.04.003.

[19] Wu, G.C.-H. and Chao, Y.-C.J. 2015. Learners' agency in a Facebook-mediated community. *Critical CALL – Proceedings of the 2015 EUROCALL Conference, Padova, Italy* (Dec. 2015), 558–563.

[20] Xue, J. et al. 2020. Public discourse and sentiment during the COVID 19 pandemic: Using Latent Dirichlet Allocation for topic modeling on Twitter. *PLOS ONE*. 15, 9 (Sep. 2020), e0239441. DOI:https://doi.org/10.1371/journal.pone.0239441.

[21] Zamani, M. et al. 2020. Understanding Weekly COVID-19 Concerns through Dynamic Content-Specific LDA Topic Modeling. *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science* (Stroudsburg, PA, USA, 2020), 193–198.

Timing Matters: Inferring Educational Twitter Community Switching from Membership Characteristics

Conrad Borchers
Carnegie Mellon University
cborcher@cs.cmu.edu

Lennart Klein
University of Tübingen
lennart.klein@student.uni-tuebingen.de

Hayden Johnson
University of Minnesota, Twin Cities
joh18320@umn.edu

Christian Fischer
University of Tübingen
christian.fischer@uni-tuebingen.de

ABSTRACT

Twitter communities receive increased attention as informal environments for teacher professional development. However, the diversity and temporal evolution in user adoption, switching, and retention are understudied. This study uses the diffusion of innovation (DOI) framework to examine user switching patterns in two large educational Twitter communities (N = 2,039,260 tweets, N = 104,847 unique users). We find that users' DOI types relate to user switching over time. Features of the DOI user classifications and other user-level characteristics explain 79% of the variance in user switching decisions and 15% of the timing of user switching. In particular, community switching depends on users' community entry time, Twitter account age, community interaction centrality, and user type (i.e., teacher vs. non-teacher). Therefore, users' perceived community fit and their relative motivation to seek out novel communities to engage with can help explain community-switching behavior. Overall, this study informs community-level interventions for user retention and a better understanding of user diversity in informal educational communities on social media.

Keywords

learner retention, social media, online communities, network analysis, diffusion of innovation, dropout

1. INTRODUCTION

Informal online learning receives increased attention in educational data mining [29, 25, 41]. Learning on social media often takes place in communities; on Twitter, hashtags connect thousands of teachers [10]. As this provides meaningful learning opportunities for teacher professional development (PD), prior research argued that social media could augment traditional PD activities [19, 24, 21].

User retention has been a topic of research investigating motivational differences for online community participation [14], social presence [26], and affiliation of central users with the community [42]. Prior work showed a gradual user dropout of the #EdChat Twitter community, with mixed success in explaining user retention through user-level features [46]. Prior research identified several user-level factors that may contribute to user retention and user membership switching in online communities outside of social media [2, 27]. These features relate to Diffusion of Innovation theory [47], network centrality [8], and social presence [36]. Yet, user community retention and switching in informal educational learning communities are understudied.

This study bridges two lines of research: First, prior findings on educational Twitter communities have focused on single-community retention and engagement [46, 42]. Second, community retention studies in educational data mining have been restricted to more formal learning settings, including MOOCs [8] and higher education [1]. Studying retention and user switching in educational Twitter communities may inform learner retention in informal online learning.

We study user retention by comparing two large and structurally different educational Twitter communities: (a) a chat-based education community and (b) a more informal "teacher lounge" community. Prior work indicates that the chat-based community's decline coincided with the lounge community's growth [20].

This study has three main contributions: First, we provide initial evidence for identifying and developing relevant features related to community retention in educational Twitter communities. Second, we compare the importance of different features in these models. This may support subsequent intervention studies and early-warning capabilities for user retention in informal learning communities. Third, we advance existing research on multi-community membership, user switching, and retention [44, 30, 28, 45]. Compared to prior work on teacher PD on Twitter, which emphasized cognitive and interactive over social-transactional tweets to bolster community retention [9, 46], we find that social integration and users' perceived community fit may matter most for community retention.

C. Borchers, L. Klein, H. Johnson, and C. Fischer. Timing matters: Inferring educational twitter community switching from membership characteristics. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 438–444, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115752

2. RESEARCH BACKGROUND

2.1 Diffusion of Innovation

Diffusion of Innovation (DOI) theory describes the adoption process of an idea or product by members of a social group [40]. Diffusion describes the adoption rate for innovation as mediated through communication. The framework consists of four major components: (a) innovation, (b) communication channels, (c) the social system, and (d) time.

DOI theory classifies members into "adopter types" [40]. These adopter types include innovators, early adopters, early majority, late majority, and laggards. Membership of an adopter type is determined by the time of adoption during the diffusion process and is often considered predictive of different behavior within the diffusion process [40]. Quantitative modeling suggests that generalizations of DOI theory can have sufficient predictive power [6].

DOI has been used prominently to explain and predict innovations in the context of social media [37, 35, 22]. Studying Twitter, prior research has used DOI to investigate factors mediating adoption by various social and political groups [39, 5, 4, 34]. This approach was extended to study the continued or discontinued use of Twitter after its initial adoption using a combination of DOI and the "uses and gratifications theory" [16, 14, 15]. Notably, prior work generally overlooks the complexities of competition between communities. This work addresses this gap by using DOI to investigate community switching behavior.

2.2 Multi-group Membership and Switching

Multi-community membership refers to users simultaneously engaging with two or more distinct user groups, with these communities typically situated in related domains. Multi-community membership can be synergistic and facilitate between-group connections. On Twitter, exchanges across hashtagged communities could form and reform multi-directional connections, effectively bridging communities and leveraging their resources and audiences [30]. Cross-community ties can also emerge due to users switching networks and retaining links to their old community [45].

Multi-community membership can facilitate between-group competition, affecting community growth and thriving [44]. This suggests that larger and older groups may experience difficulty growing their membership and are more vulnerable to competitive pressure. At the same time, community members identify community leaders via sociability, knowledge contribution behaviors, and structural social capital (often operationalized via betweenness centrality; [18]). Therefore, influential users play a central role in the competition between established and emerging communities.

Connections in emerging communities are based on geographic and social similarities between users. For example, the types of social ties in communities are often correlated with community age [28]. In established communities, sharing is more predicated on expertise. Moreover, prior work found user membership characteristics (e.g., engagement rate, professional role, and user account age) to relate to the sentiment in educational Twitter communities [41]. Given these systematic differences in online communities, research may exploit these variations to predict community membership. Prior work on such predictive models is scarce but indicated that models might predict community membership via social ties, user attribute homophily, and existing community memberships for a set of game-based communities [3].

2.3 The Present Study

This study investigates two large educational Twitter communities with considerable user and time overlap regarding user activity. Prior research on user community switching and retention primarily focused on non-educational communities [44, 30, 28, 45] and put limited focus on explaining the determinants of user switching based on community- and user-level features. Understanding the determinants of user switching and its timing offers novel lenses into user switching behavior and opens up the potential to intervene and retain users in educational communities. We investigate the following three research questions (RQs):

RQ1: How did the user base of two large educational Twitter communities overlap over time according to the Diffusion of Innovation model?

RQ2: How can user switching between two large educational Twitter communities be inferred using community membership and user-level features?

RQ3: How can the relative timing of user switches between two large educational Twitter communities be inferred using community membership and user-level features?

3. METHOD

3.1 Sample Description

This study uses data from a large project that examined the entire German educational Twittersphere [20]. The corresponding data download occurred between April 8 - 25, 2022, with the Twitter API 2.1. Our sample includes all tweets until the end of 2021 in Germany's two largest educational Twitter communities: the EdChatDE and the TWLZ community.

EdChatDE is the German chapter of the American EdChat network, which holds and facilitates regular chat hours for educators. In contrast, the TWLZ (an abbreviation for "Twitterlehrerzimmer," which translates to "Twitter teacher's lounge") is an umbrella community for education professionals to talk, connect, and share content across subject areas and school levels.

Notably, our data does not only include tweets that used the sampled hashtags but also all of their conversation tweets (i.e., replies to the tweets, including the respective hashtags). We removed 255,061 tweets from 170 identified bot accounts. Bot detection followed a hybrid approach based on keyword filtering in user bios and human coding as described in [20]. This led to a full study sample of 2,039,260 tweets from 104,847 unique users.

3.2 Measures

This study infers user switching and timing from user roles based on the DOI theory and other features mined from Twitter data. The code of our analyses is publicly available.[1]

[1] github.com/conradborchers/community-switch-edm23

Community membership and switching. We defined community membership as users having at least two community interactions (i.e., mentioning, quoting, replying, or retweeting another user in a post that contributed to the community). This number was determined by investigating a logistic growth model's fit via RSS (as assumed by the DOI model). We checked the robustness of our results across one, two, and three required interactions.

Notably, the last 10% of user interactions are not counted as community interactions. This hedges against artificially prolonging users' community membership beyond a long break through random postings long after their primary community engagement ended. We verified the robustness of our results against a 15% and 5% cutoff.

We define user switching as the point in time when users become members of one community and cease to be members of another community (Equation 1). Our definition allows for a time gap where users are members of neither community before joining the community to which they switch.

$$t_{switch[a \to b]} = max(t_{exit[a]}, t_{entry[b]}) \mid t_{exit[b]} > t_{exit[a]} \quad (1)$$

DOI user types. We classified users based on the time they joined the chat community following DOI theory [32]. Users are ranked based on how early they joined a community, with groups separated by quantiles. This results in the following variable levels: innovators (earliest 2.5%), early adopters (13.5%), early majority (34%), late majority (34%), and laggards (remaining 16%).

Social network analysis. To operationalize the social status of users within a community, we calculate a set of common centrality measures, including degree, closeness, betweenness, and eigenvector centrality on an unweighted and undirected network of user-to-user interactions [7]. *Degree centrality* refers to the number of nodes (i.e., other users) a user has interacted with. *Closeness centrality* measures the distance between a node and every other node in the network. That means well-connected users have shorter connection paths to all other users. *Betweenness centrality* is a measure of a node's function as a bridge to connect other nodes. *Eigenvector centrality* describes the importance of a node by the sum of the centrality of the nodes it connects to. Thus, a user interacting with a central user gains importance in the network.

Teacher classification. We trained a supervised learning classifier based on a training data set of 1,000 randomly-sampled user profiles [38]. Two experienced human coders first labeled users' Twitter bios and up to 50 sample tweets into "teachers" and "non-teachers." Their inter-rater reliability was substantial with $\kappa = 0.77$ [12]. Subsequently, we trained and tested multiple text-based teacher classification models using different algorithms, from which a logistic regression model emerged as the most predictive. The model was optimized with hyperparameter tuning via grid search and 10-fold cross-validation and achieved a holdout test set ($N = 250$) accuracy of $AUC = 0.79$. Applied to our study sample, 10,313 users were classified as teachers, contributing 1,078,976 tweets while 94,534 non-teachers contributed 960,284 tweets.

Twitter engagement. User and social engagement variables are provided by the Twitter API, including a continuous variable on the user's lifespan, that is, the time passed (in days) since they first joined Twitter. Also, we included two continuous variables indicating the number of followers and followings. This allows us to gauge a user's popularity and connectedness on the platform. Lastly, we included two continuous variables describing users' posting behavior: the average number of tweets a user posted per day and their total number of reposts (i.e., retweets and quotes). These measures indicate a user's level of engagement and responsiveness.

3.3 Analytical Methods

RQ1 reports descriptive statistics on the size and overlap of two large educational Twitter communities. Then, we describe the descriptive overlap in DOI membership types between these two communities for active users.

RQ2 applies logistic regression models to infer user switching behavior. We employ AIC-based backward search to determine a parsimonious and interpretable user switching model [31]. We z-standardized all numeric variables to aid model coefficient interpretations. Additionally, we log-transformed network centrality measures before standardizing them, given their heavy-tailed distributions [17]. For all linear models, we verified that modeling assumptions (e.g., normal distribution of residuals, homoscedasticity, and linearity assumptions) are not violated through inspection of corresponding diagnostic plots.

RQ3 replicates the modeling procedure presented in RQ2 using the z-standardized relative timing of user switching as the dependent variable with ordinary least squares (OLS) regression models.

4. RESULTS

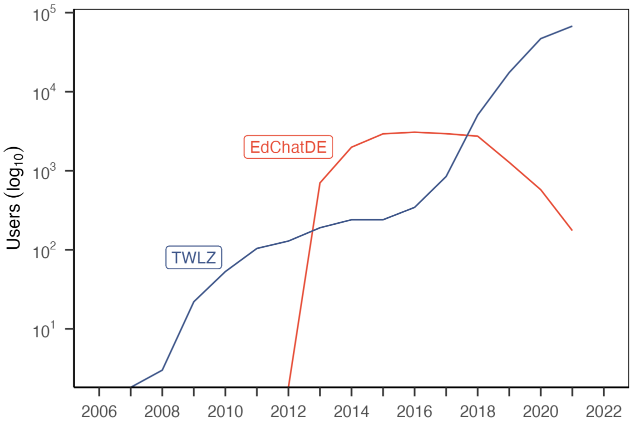
4.1 Community User Overlap (RQ1)

The chat community included N = 5,391 members, while the lounge community included N = 69,877 members. N = 2,775 users had dual membership for an average of $M = 243$ days ($SD = 273$ days). The median number of active days was 1,273 for the chat community and 317 for the lounge community. We report the longitudinal development of user numbers in both communities in Figure 1. Most notably, the decline of user numbers in the chat-based community occurred shortly after the lounge community experienced exponential user growth starting in 2017.

We found that N = 2,891 (65.98%) of chat community members switched to the Twitter lounge. The median switching time was 842 days after joining the chat-based community with an IRQ of 1,218 days, yielding a considerable variance in whether users switched and their exact time of switching. Associations between the DOI user types of both communities are displayed in Figure 2.

Figure 2 indicates that chat community laggards were often early adopters of the teacher lounge community. Notably, these users were also less often innovators in the teacher lounge community. Conversely, the late majority of the chat-based community were more often innovators and less often early adopters of the teacher lounge community. A χ^2 independence test rejected the independence of both user classifications ($\chi^2 = 320.82$, $df = 16$, $p < .001$).

Figure 1: Community development over time, including unique user counts per year, transformed to log scale to represent growth rates.



4.2 Determinants of Community Switching (RQ2)

We investigate user switching determinants from the chat community to the teacher lounge community via logistic regression (Table 1). The four main findings are as follows:

First, the later users joined the chat community, the more likely they were to switch to the teacher lounge community. Effect sizes ranged between $OR = 1.10$ ($p = .863$) for early adopters compared to innovators and $OR = 296.87$ ($p < .001$) for early adopters compared to innovators. Second, users whose Twitter account was a standard deviation older than average were around 24 times more likely to switch ($OR = 24.49$, $p < .001$). Third, teachers were more than three times as likely to switch to the teacher lounge community than non-teachers ($OR = 3.06$, $p < .001$). Fourth, users with larger followings and more followed accounts were less likely to switch ($OR = 0.76$, $p < .001$ and $OR = 0.60$, $p < .001$, respectively). Fourth, among our centrality measures, degree centrality exhibited the largest effect size and was positively associated with user switching ($OR = 2.48$, $p < .001$).

4.3 Timing of Community Switching (RQ3)

Analogous to RQ2, we investigate the determinants of the relative timing of user switching using ordinary least squares regression (Table 2). The three main findings are as follows:

First, users that joined the chat community later were also more likely to join the teacher lounge community later. While early adopters joined the teacher lounge community $\beta = 0.13$ standard deviations (approx. 50 days) later ($p = .055$) compared to innovators, laggards joined the teacher lounge community $\beta = 0.42$ standard deviations (approx. 163 days) later ($p < .001$) compared to innovators. Second, teachers switched $\beta = -0.29$ standard deviations (approx. 112 days) earlier ($p < .001$) than non-teachers. Third, degree centrality was most strongly associated with the relative timing of user switching. Per additional standard deviation in user degree centrality, users switched $\beta = -0.37$ standard deviations (approx. 143 days) earlier ($p < .001$).

Figure 2: Crosstable of DOI adopter categories with colored residuals from the expected frequencies of a χ^2 test.

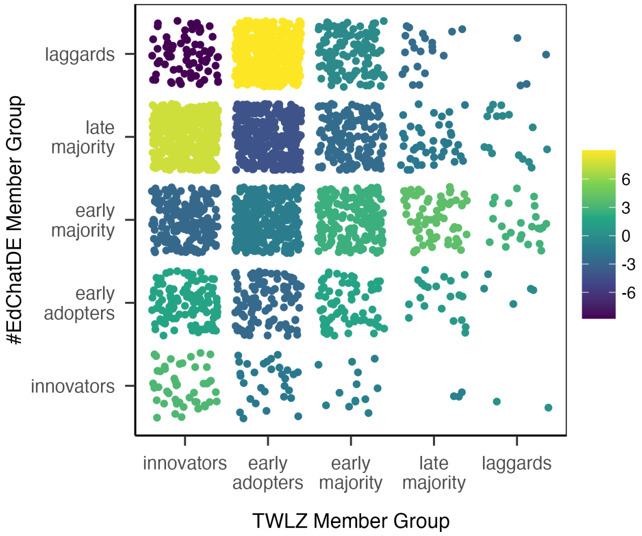


Table 1: Logistic regression model on user switching behavior.

Effect	OR	SE
(Intercept)	0.05***	0.51
Chat DOI group [vs. innovators]		
Early adopters	1.10	0.53
Early majority	9.34***	0.51
Late majority	60.10***	0.53
Laggards	296.87***	0.55
Teacher [vs. non-teacher]	3.06***	0.18
Lifespan (days)	24.49***	0.11
User following	0.60***	0.09
User followers	0.76***	0.08
Chat tweets per day	0.76***	0.06
Degree Centrality	2.48***	0.14
Closeness Centrality	0.63***	0.07
Betweenness Centrality	1.61***	0.10
Observations 5,391		
R^2 Tjur 0.785		

* $p < .05$, ** $p < .01$, *** $p < .001$

User classification features had among the largest effect sizes. Figure 3 illustrates interactions between DOI and teacher user classifications to further examine associations of DOI types across different user types. Notably, teachers consistently switched communities across all DOI user types earlier than non-teachers. However, the difference between teachers and non-teachers diminished the later users switched. In particular, the smallest difference in medians between teachers and non-teachers was for the late majority (0.12 SD; approx. 48 days) and laggards (0.43 SD; approx. 166 days) of the chat-based community.

Table 2: OLS regression on the relative user switch time.

Effect	β	SE
(Intercept)	-0.01	0.11
Chat DOI group [vs. innovators]		
Early adopters	0.14	0.11
Early majority	0.18	0.11
Late majority	-0.06	0.12
Laggards	0.43**	0.13
Teacher [vs. non-teacher]	-0.29***	0.04
Lifespan (days)	-0.08**	0.03
User following	0.04*	0.02
User reposts	-0.04*	0.02
Degree Centrality	-0.37***	0.04
Closeness Centrality	-0.17***	0.04
Betweenness Centrality	0.16***	0.03
Eigenvector Centrality	0.05	0.04
Observations	2,693	
R^2 / Adjusted R^2	0.145 / 0.141	

* $p < .05$, ** $p < .01$, *** $p < .001$

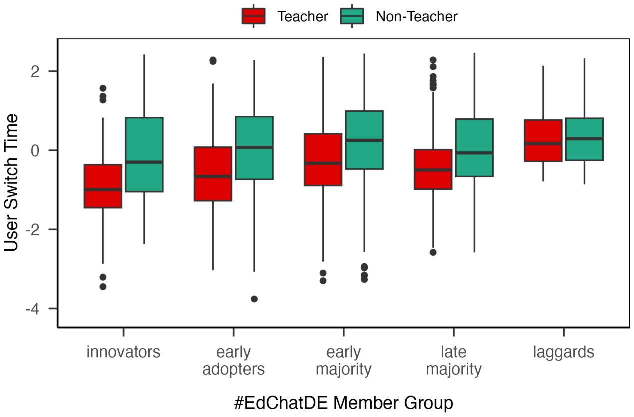
5. DISCUSSION

This study examines user switching determinants between two large education-related Twitter communities through a DOI theory lens. While prior studies focused on the growth and retention in single communities [16, 14, 15], less attention has been given to user characteristics explaining community switching. Our three main findings are as follows:

First, community switching was more likely the longer users were active on Twitter and the later they joined the community from which they switched. This is important as prior work focused on increasing cognitive and interactive tweets over social-transactional tweets to increase retention in PD communities [9, 46]. However, our findings suggest that social integration into the community matters for community retention. Users that joined a community late may have more challenges in making connections or building a reputation in the community. Therefore, they might be more likely to switch communities. Alternatively, the entry barrier may feel higher for newcomers in more established communities, given that sharing in older communities tends to be predicated on expertise rather than social and geographic similarity [28]. Notably, we found the relative community joining time measures to have large effect sizes encouraging future research to use relative community join times as measures for understanding informal learning communities.

Second, teachers were around three times more likely to switch communities and switched around 112 days earlier compared to non-teachers. Given that the community to which users switched in our sample is an "Twitter teacher lounge," these observed effects may relate to an increased perceived fit for teachers regarding the target community [11]. This interpretation aligns with findings from marketing research investigating customer adoption of products moderated by self-identity [13, 43]. Therefore, communities may improve user adoption by explicitly addressing their target audiences. Future research may investigate more features of perceived role fit to infer educational community retention.

Figure 3: Group-wise box plots based on DOI types and teacher classification for the standardized user switching time. One SD equals approximately 386 days.



Third, central community members were more likely to switch communities. This finding extends prior work suggesting that interactions with highly influential users positively relate to community retention [42] and prolonged engagement in blogs [33]. A potential interpretation is that highly central users may have a stronger tendency to adopt and seek novel participation opportunities in educational Twitter, irrespective of their integration into the community. Alternatively, the decline of the chat-based community in our data set may be predicated on highly influential users, being the community backbones, leaving the community. Comparing both explanations in future research may improve community intervention efforts, for example, by targeting highly central users in the network to continue their engagement and boost community health and longevity.

5.1 Limitations and Future Work

This study only examined two Twitter communities. Specific events, such as the COVID-19 pandemic, may have influenced community growth, during which the teacher lounge community experienced large growth [23]. In addition, our data are correlational and do not allow for causal claims. However, future work may leverage longitudinal or hierarchical modeling with repeated user engagement measurements to better understand the strong association of DOI groups with user switching. Similarly, future work may explore user-level differences in device usage (e.g., desktop vs. mobile use) or create an early warning system flagging declining engagement of central users.

Taken together, our findings can provide important insights for stakeholders initiating, studying, and orchestrating informal PD spaces on Twitter and inform research on informal learning in digital spaces more broadly.

6. REFERENCES

[1] F. Agrusti, G. Bonavolontà, and M. Mezzini. University dropout prediction through educational data mining techniques: A systematic review. *Journal of E-Learning and Knowledge Society*, 15(3):161–182,

2019.

[2] T. Althoff and J. Leskovec. Donor retention in online crowdfunding communities: A case study of donorschoose. org. In *Proceedings of the 24th International Conference on World Wide Web*, pages 34–44, 2015.

[3] H. Alvari, K. Lakkaraju, G. Sukthankar, and J. Whetzel. Predicting guild membership in massively multiplayer online games. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*, pages 215–222. Springer, 2014.

[4] M. Anderson, K. Lewis, and O. Dedehayir. Diffusion of innovation in the public sector: Twitter adoption by municipal police departments in the us. In *2015 Portland International Conference on Management of Engineering and Technology (PICMET)*, pages 2453–2464. IEEE, 2015.

[5] M. M. Archibald and A. M. Clark. Twitter and nursing research: how diffusion of innovation theory can help uptake. *Journal of Advanced Nursing*, 70(3):e3–e5, 2014.

[6] F. M. Bass. A new product growth for model consumer durables. *Management Science*, 15(5):215–227, 1969.

[7] F. Bloch, M. O. Jackson, and P. Tebaldi. Centrality Measures in Networks. (arXiv:1608.05845), 2021.

[8] A. Bozkurt. Surfing on three waves of MOOCs: An examination and snapshot of research in massive open online courses. *Open Praxis*, 13(3):296–311, 2021.

[9] J. Carpenter and D. Krutka. Engagement through microblogging: Educator professional development via Twitter. *Professional Development in Education*, 41(4):707–728, 2015.

[10] J. P. Carpenter and S. A. Morrison. Enhancing teacher education... with Twitter? *Phi Delta Kappan*, 100(1):25–28, 2018.

[11] S. M. Chan-Olmsted, M. Cho, and S. Lee. User perceptions of social media: A comparative study of perceived characteristics and user profiles by social media. *Online Journal of Communication and Media Technologies*, 3(4):149–178, 2013.

[12] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[13] I. Confente, D. Scarpi, and I. Russo. Marketing a new generation of bio-plastics products for a circular economy: The role of green self-identity, self-congruity, and perceived value. *Journal of Business Research*, 112:431–439, 2020.

[14] K. Constantinos and Y. Coursaris. Understanding Twitter's adoption and use continuance: The synergy between uses and gratifications and diffusion of innovations. *SIGHCI 2010 Proceedings. Paper*, 3, 2010.

[15] C. K. Coursaris, W. Van Osch, J. Sung, and Y. Yun. Disentangling Twitter's adoption and use (dis) continuance: A theoretical and empirical amalgamation of uses and gratifications and diffusion of innovations. *AIS Transactions on Human-Computer Interaction*, 5(1):57–83, 2013.

[16] C. K. Coursaris, Y. Yun, and J. Sung. Twitter Users vs. Quitters: a Uses and Gratifications and Diffusion of Innovations approach in understanding the role of mobility in microblogging. In *2010 Ninth International Conference on Mobile Business and 2010 Ninth Global Mobility Roundtable (ICMB-GMR)*, pages 481–486. IEEE, 2010.

[17] K. Dasaratha. Distributions of centrality on networks. *Games and Economic Behavior*, 122:1–27, 2020.

[18] S. Faraj, S. Kudaravalli, and M. Wasko. Leading collaboration in online communities. *MIS quarterly*, 39(2):393–412, 2015.

[19] C. Fischer, B. Fishman, and S. Y. Schoenebeck. New contexts for professional learning: Analyzing high school science teachers' engagement on Twitter. *AERA Open*, 5(4):1–20, 2019.

[20] C. Fischer, L. Klein, C. Borchers, and F. Morina. Mapping the landscape of educational Twitter use in Germany: Informal teacher learning in online communities of practice. *OSF Preprints*, 2023.

[21] C. Fischer, Y. Omarchevska, T. Fütterer, and J. Rosenberg. How do teachers collaborate in informal professional learning activities? An epistemic network analysis. *OSF Preprints*, 2022.

[22] O. Folorunso, R. O. Vincent, A. F. Adekoya, and A. O. Ogunde. Diffusion of innovation in social networking sites among university students. *International Journal of Computer Science and Security*, 4(3):361–372, 2010.

[23] T. Fütterer, E. Hoch, K. Stürmer, A. Lachner, C. Fischer, and K. Scheiter. Was bewegt Lehrpersonen während der Schulschließungen? Eine Analyse der Kommunikation im Twitter-Lehrerzimmer über Chancen und Herausforderungen digitalen Unterrichts. *Zeitschrift für Erziehungswissenschaft*, 24(2):443–477, 2021.

[24] C. Greenhow, S. M. Galvin, D. L. Brandon, and E. Askari. A decade of research on K–12 teaching and teacher learning with social media: Insights on the state of the field. *Teachers College Record*, 122(6):1–72, 2020.

[25] A. Gruzd, D. Paulin, and C. Haythornthwaite. Analyzing social media and learning through content and social network analysis: A faceted methodological approach. *Journal of Learning Analytics*, 3(3):46–71, 2016.

[26] S. Han, J. Min, and H. Lee. Antecedents of social presence and gratification of social connection needs in SNS: a study of Twitter users and their mobile and non-mobile usage. *International Journal of Information Management*, 35(4):459–471, 2015.

[27] K. S. Hone and G. R. El Said. Exploring the factors affecting MOOC retention: A survey study. *Computers & Education*, 98:157–168, 2016.

[28] E. H. Hwang, P. V. Singh, and L. Argote. Knowledge sharing in online communities: Learning to cross geographic and hierarchical boundaries. *Organization Science*, 26(6):1593–1611, 2015.

[29] S. Joksimović, V. Kovanović, J. Jovanović, A. Zouaq, D. Gašević, and M. Hatala. What do cMOOC participants talk about in social media? a topic analysis of discourse in a cMOOC. In *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge*, pages 156–165, 2015.

[30] J. Jones. Switching in Twitter's hashtagged exchanges. *Journal of Business and Technical Communication*, 28(1):83–108, 2014.

[31] A. Jović, K. Brkić, and N. Bogunović. A review of feature selection methods with applications. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1200–1205. IEEE, 2015.

[32] J. Kaminski. Diffusion of innovation theory. *Canadian Journal of Nursing Informatics*, 6(2):1–6, 2011.

[33] I. Kayes and J. Chakareski. Retention in online blogging: a case study of the blogster community. *IEEE Transactions on Computational Social Systems*, 2(1):1–14, 2015.

[34] C. Kim and S. Lee. Innovation vs. normalization: Politicians' Twitter use at the early majority stage of its diffusion in the Korean assembly. *The Social Science Journal*, pages 1–13, 2020.

[35] A. Leerapong. Applying diffusion of innovation in online purchase intention through social network: A focus group study of Facebook in Thailand. *Information Management and Business Review*, 5(3):144–154, 2013.

[36] S. Y. Liu, J. Gomez, and C.-J. Yen. Community college online course retention and final grade: Predictability of social presence. *Journal of Interactive Online Learning*, 8(2), 2009.

[37] L. Ma, C. S. Lee, and D. H.-L. Goh. Understanding news sharing in social media: An explanation from the diffusion of innovations theory. *Online Information Review*, 38(5):598–615, 2014.

[38] M. Pennacchiotti and A.-M. Popescu. A Machine Learning Approach to Twitter User Classification. *Proceedings of the International AAAI Conference on Web and Social Media*, 5(1):281–288, 2011.

[39] R. D. Peterson. To tweet or not to tweet: Exploring the determinants of early adoption of Twitter by House members in the 111th Congress. *The Social Science Journal*, 49(4):430–438, 2012.

[40] E. M. Rogers, A. Singhal, and M. M. Quinlan. Diffusion of innovations. In *An Integrated Approach to Communication Theory and Research*, pages 432–448. Routledge, 2014.

[41] J. M. Rosenberg, C. Borchers, E. B. Dyer, D. Anderson, and C. Fischer. Understanding public sentiment about educational reforms: The next generation science standards on Twitter. *AERA open*, 7:1–17, 2021.

[42] J. M. Rosenberg, J. W. Reid, E. B. Dyer, M. J. Koehler, C. Fischer, and T. J. McKenna. Idle chatter or compelling conversation? the potential of the social media-based #NGSSchat network for supporting science education reform efforts. *Journal of Research in Science Teaching*, 57(9):1322–1355, 2020.

[43] A. P. Schouten, L. Janssen, and M. Verspaget. Celebrity vs. influencer endorsements in advertising: the role of identification, credibility, and product-endorser fit. *International journal of advertising*, 39(2):258–281, 2020.

[44] X. Wang, B. S. Butler, and Y. Ren. The impact of membership overlap on growth: An ecological competition view of online groups. *Organization Science*, 24(2):414–431, 2013.

[45] Y. Wei, W. Zhang, S. Yang, and X. Chen. Online social network with communities: Evolvement of within-and cross-community ties. *Available at SSRN 3420525*, 2021.

[46] W. Xing and F. Gao. Exploring the relationship between online discourse and commitment in Twitter professional learning communities. *Computers & Education*, 126:388–398, 2018.

[47] M. Zhu. Analysis of technology innovation diffusion model and diffusion characteristics on educational technology: Case study of MOOC. In *Proceedings of the 2022 5th International Conference on Mathematics and Statistics*, pages 47–51, 2022.

Understanding Revision Behavior in Adaptive Writing Support Systems for Education

Luca Mouchel
EPFL
luca.mouchel@epfl.ch

Thiemo Wambsganss
EPFL
thiemo.wambsganss@epfl.ch

Paola Mejia-Domenzain
EPFL
paola.mejia@epfl.ch

Tanja Käser
EPFL
tanja.kaeser@epfl.ch

ABSTRACT

Revision behavior in adaptive writing support systems is an important and relatively new area of research that can improve the design and effectiveness of these tools, and promote students' self-regulated learning (SRL). Understanding how these tools are used is key to improving them to better support learners in their writing and learning processes. In this paper, we present a novel pipeline with insights into the revision behavior of students at scale. We leverage a data set of two groups using an adaptive writing support tool in an educational setting. With our novel pipeline, we show that the tool was effective in promoting revision among the learners. Depending on the writing feedback, we were able to analyze different strategies of learners when revising their texts, we found that users of the exemplary case improved over time and that females tend to be more efficient. Our research contributes a pipeline for measuring SRL behaviors at scale in writing tasks (i.e., engagement or revision behavior) and informs the design of future adaptive writing support systems for education, with the goal of enhancing their effectiveness in supporting student writing. The source code is available at `https://github.com/lucamouchel/Understanding-Revision-Behavior`.

Keywords

Revision Behavior, Writing Support Systems, ML-based adaptive feedback, Self-Regulated Learning

1. INTRODUCTION

Intelligent writing support tools (e.g., Grammarly, WordTune or Quilbot) offer new ways for learners to receive feedback and thus revise their texts [19]. These and other writing support systems bear the potential to provide learners with needed adaptive feedback on their writing exercises when educators are not present, (e.g., on grammatical mistakes [32], argumentation [26], empathy [29], or general persuasive writing [27]). They can help students in their self-regulated learning (SRL) process [11, 36], to organize their thoughts and ideas, reflect on their learnings, or simply receive feedback on frequently occurring grammar or argumentation mistakes. From an educational perspective, it is important to understand how these tools are used by learners in educational settings and how they improve the effectiveness of educational scenarios [7, 18]. Present research is largely focused on designing and building writing support systems [8, 20]. However, there are not many insights into the effects of the usage of these tools and their impact on students' SRL processes [4, 25], which is why we contribute a novel pipeline analyzing and visualizing revision behavior to better understand how we can design, develop and improve existing systems to better support students. Techniques from the field of data mining are a solution to understanding revision behavior and explaining SRL. One such technique is Keystroke Logging (KL). KL allows us to use educational data mining to analyze user behavior in writing tasks [16, 35, 23][1]. In this study, we model, inspect, and analyze quantitative data in learners' writing interactions through KL by developing a novel pipeline. We use a keystroke log from an experiment, where users were divided into two groups. The first one was given adaptive feedback and the second one was not. A detailed description of our dataset and the experiment demographics and procedure are available in Section 3. To the best of our knowledge, no publicly available pipeline exists that focuses on processing the keystroke behavior of learners and helps analyze SRL characteristics such as engagement, revision, or visualize the learning path. We intend to first identify and visualize the differences between these two groups in their revision process and compare different user profiles and measure their engagement over time. We use an exemplary data set to build this pipeline and apply data mining in order to gain insights into the underlying process of this writing activity.

2. BACKGROUND

Research on Automatic Data Mining for Writing Behaviour

Research in writing process analysis can be traced to the 1970s [9, 24]. However, only more recently have studies been focusing theoretically on behavioral and cognitive processes

L. Mouchel, T. Wambsganss, P. Mejia-Domenzain, and T. Käser. Understanding revision behavior in adaptive writing support systems for education. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 445–452, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115766

[1] Tools such as InputLog [16] or ETS [35] are examples of KL programs.

of writing [14, 17]. In fact, Flower and Hayes [10] laid the groundwork for research on the psychology of writing. They propose that the act of writing is propelled by goals, which are created by the writer and grow in number as the writing progresses. Today, writing support tools need to support this cognitive process as it emphasizes writers' intentions, rather than their actions [12]. It is important to understand what these tools help with, and how we may design new ones [12]. While prior works on text revision [8, 15, 20, 28] have proposed machine collaborative writing interfaces, they focus on collecting human-machine interaction data to better train neural models, rather than understanding the underlying processes of text revision. Several studies in the past have used KL as a technique to study revision [16, 23, 35] in different settings and some of the aims were to understand and evaluate keystroke log features in a writing task context. However, until now, KL has been scarcely used in the classroom [25]. One issue with keystroke loggers is their invasive nature. KL raises several ethical issues, most notably privacy violation [25], but in this study, participants gave their consent for the collection of their data, all the while preserving their privacy. Previous research has suggested that writing time and number of keystrokes, which are indicative of general writing fluency and effort, are related to writing quality [1, 33]. Another feature of interest is pause times, [34] found that under a certain timed-writing test condition, shorter pauses are preferred as that indicates an adequate understanding of the task requirements, more familiarity with the writing topic, and better task planning [23].

Self-Regulated Learning

To analyze revision behavior, we rely on the lens of self-regulated learning (SRL). SRL refers to the pro-active process that learners engage in to optimize their learning outcome [36]. According to Zimmerman's model of SRL [36], there are three major phases: forethought, performance and self-reflection. The forethought phase includes task analysis, such as goal setting and strategic planning and self-motivational beliefs. The performance phase includes self-control processes, such as task and attention-focusing strategies. The self-reflection phase includes processes involving self-judgment and self-reaction [31]. SRL is essential in the context of studying revision behavior in writing support systems as it allows writers to take an active role in identifying and addressing their own writing weaknesses, rather than simply relying on the writing support system to automatically detect and correct errors. This can lead to a deeper understanding of the writing process.

3. METHOD

To investigate revision behavior in the writing process, we propose a pipeline for the automatic analysis of the SRL behavior of users during a writing task. Our work follows the Knowledge Discovery in Databases process by following the methodology in Fig. 1.

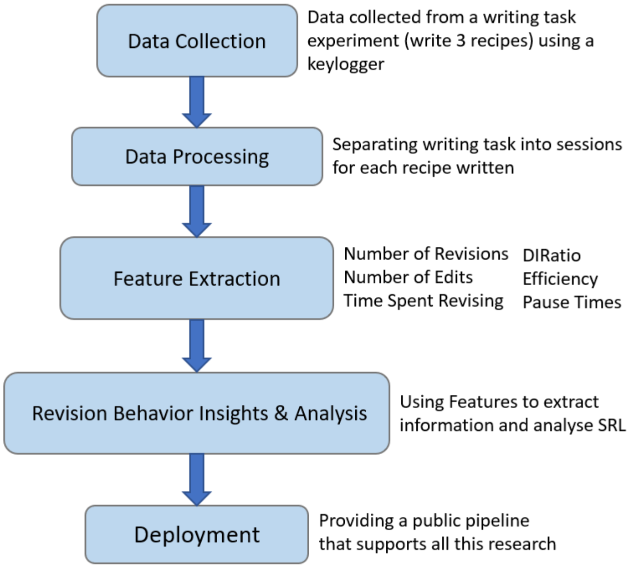

Data Collection
Data collected from a writing task experiment (write 3 recipes) using a keylogger
Data Processing
Separating writing task into sessions for each recipe written
Feature Extraction
Number of Revisions
Number of Edits
Time Spent Revising
DIRatio
Efficiency
Pause Times
Revision Behavior Insights & Analysis
Using Features to extract information and analyse SRL
Deployment
Providing a public pipeline that supports all this research

Figure 1: Overview of our pipeline and methodology, following the KDD process

Demographics, Procedure & Dataset Description

With approval of the ethical board of our university, we collected data from a writing experiment which consisted of 73 users divided into two groups, as illustrated in Table 1.

Table 1: Demographics of the participants per group from the exemplary data set

	With Adaptive Feedback (G1)	Without Adaptive Feedback (G2)
No. Participants	34	39
Age Mean	26.8	26.3
Age Std	3.3	2.8
% Female	43	51
% Male	51	46
% Other	6	3

The two groups of users were tasked with writing three cooking recipes. Both groups were given a sample recipe as reference. The first group (G1) received adaptive feedback from the platform when they submitted their texts. The second group (G2) did not receive any feedback. Once they submitted their recipes to the system, users in G1 had the option to reset and start a new recipe or revise their texts based on the feedback. The same protocol was followed for G2, but they did not receive feedback. Here are several examples of the feedback the platform provided users: *'List each ingredient separately.'*, *'Enumerate the steps.'*, *'How can your recipe be more specific?'*, *'Use stir, mix, or beat instead of "add" to be more specific.'* or *'Indicate whether the meat, poultry, or seafood is boned, skinned, or otherwise prepared.'*

With regards to the dataset, the entries of the log data we collected consisted of user ids, event dates, the keystroke logs as a JSON file and the final version of the text submitted at that particular date. An example of an entry is as follows: `2023-01-01, 12:00:00, user1, [{'time': 1, 'character': 'a'}, ... }], "a) Cook ..."`.

Qualitative Perception

Following the experiment, users were tasked with answering follow-up questions and we identified eight different topics regarding the reported revisions, including, *adding missing ingredients*, *improving the clarity*, *not making any changes* and others. To do this, we used `BERTTopic` [13], a topic modeling technique that clusters sentence embeddings generated by `Sentence-BERT` [22], to perform qualitative analysis of participants' open responses about recipe revisions: (*What did you edit (add, remove or change) from the original text (the recipe you wrote)?*). We split the sentences into clusters based on their relevance, assigned names to each cluster, and computed the probability of each sentence belonging to a cluster. We grouped the sentences by participant to obtain the set of topics associated with their entire text answer. For example, if a participant's answer consisted of sentences with assigned topics A, B, and C, the set of topics associated with their answer would be $Z = \{A, B, C\}$.

Data Processing

Given that the logs consist of the users' first attempts at writing one of the three recipes and their respective revision phases, it is important to separate them in order to focus only on the revision steps. We define sessions for a user as all the data collected from them for **one** recipe. To separate sessions, we use cosine distance to detect where the session ends and where the next one starts. One advantage of using cosine distance for text comparison is that it is relatively insensitive to the length of the strings. In contrast, other measures of distance such as Euclidean distance are sensitive to the length of the vectors and can be affected by the presence of common words that do not contribute significantly to the meaning of the strings. To map sentences to 50-dimensional vectors, we use a GloVe model [21], which is already trained on Wikipedia. First, we map each word to their embeddings and then compute the sum of the vectors component-wise. Formally, each text submitted t has a set of words $\mathcal{W} = \{w_i \mid 1 \leq i \leq N_t\}$, where N_t is the number of words for text t. Then, we map each w_i to their embeddings $\tilde{w_i}$ which are 50-dimensional vectors. Now let $\tilde{t}$ be the embedding of the text t, then $\tilde{t} = \sum_{i=1}^{N_t} \tilde{w_i}$. This allows us to capture each word of the text and this way, we can collect the set of all text embeddings in the dataset $\mathcal{T} = \{\tilde{t_1}, \tilde{t_2}, ...\} = \left\{\sum_{i=1}^{N_1} \tilde{w_i}, \sum_{i=1}^{N_2} \tilde{w_i}, ...\right\}$. We use $\mathcal{T}$ to run the recursive algorithm described in Appendix C on the recipes submitted and compute the cosine distance between the text embeddings of t_k, t_{k+1}, ..., starting at $k = 0$, until we find $n > k$

$$1 - \frac{\langle \tilde{t_k},\ \tilde{t_n} \rangle}{\|\tilde{t_k}\| \cdot \|\tilde{t_n}\|} < 0.995$$

When we do, we define n as the index of a new recipe in our data. Then, we repeat the process by starting at t_n and comparing $\tilde{t_n}$ with the text embeddings $\tilde{t}_{n+1}, \tilde{t}_{n+2}, ...$ to find the next index. This way, we collect the indices of new recipes in our dataset so that we can focus on the revision between these indices.

Moreover, to apply process mining techniques, we built event logs from the writing task. For each group, we collect the activities for each user, by looking at when they submit the first, second and third recipes and all the revision steps in between.

Feature Extraction

Different aspects of SRL have been researched extensively [18]. In a meta-analysis on online education, [6] found significant associations with academic achievement for five subscales of SRL: effort regulation (persistence in learning), time management (ability to plan study time), metacognition (awareness and control of thoughts), critical thinking (ability to carefully examine material), and help-seeking (obtaining assistance if needed)[2]. Based on these findings, we use the following dimensions to represent student behavior: effort regulation (Number of Revisions, Number of Edits, Time Spent Revising), time management (Time Spent Revising, Pause Times), metacognition (Efficiency, Pause Time), and critical thinking (DIRatio). A detailed description of these feature variables can be found in Appendix A, Table 3.

Building the Learning Path

Understanding revision behavior implies understanding the underlying process in the writing task (e.g., how long do users in a group take to revise on average or how many users revise). In order to understand this better, process mining, especially process discovery [5], can help us model and visualize the writing process for users in a group and design a learning path when using adaptive writing support systems [30]. In this study, we use Directly-Follows Graphs (DFGs) [3], which represent activities and their relationships[3]. This is useful for the field of SRL as it provides a way to visualize and analyze the steps involved in a process, especially revision. A formal definition of DFGs can be found in Appendix B.

4. RESULTS

Revision Strategies

With this study, we find that users in different groups revise their texts differently. Recall that `G1` is given adaptive feedback and `G2` is not. By providing insightful feedback on what a user can change in their writing, users tend to have more revision steps with fewer edits at each step. However, users not receiving feedback follow the opposite trend, they have fewer revision steps, with a larger number of revisions at each step. This phenomenon is visible in Fig. 2. In fact, for the first and second texts (Appendix D, Tables 4 and 5), we find p-values < 0.05 for Number of Revisions using t-tests, which indicates a significant difference in the number of revisions. This is also underlined by the mean number of revisions and edits. On average, users in `G1` tend to revise their texts more often, with fewer edits at each step (Appendix D, Tables 4 to 6). From the directly-follows graphs (Fig. 3), we see that users spend approximately the same amount of time writing recipes and the same amount of time revising at the first revision step. However, we see that users in `G2` revise much longer when having consecutive revision sessions (6 min on average) compared to `G1` (56 s)(Fig. 3). This confirms that users in `G1` have shorter revision sessions, whereas users in `G2` have longer revision steps.

[2] The nature of our log data does not allow to represent *help-seeking*.

[3] Other data structures like Petri Nets could also be used. Petri Nets are commonly used to apply process mining [3].

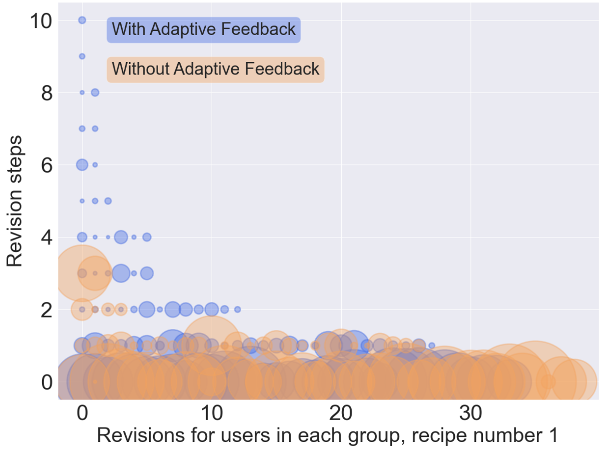


Figure 2: Bubble plot for the first recipe sorted by the number of revisions. The bubbles correspond to the number of edits (insertions and deletions) for a user at each revision step

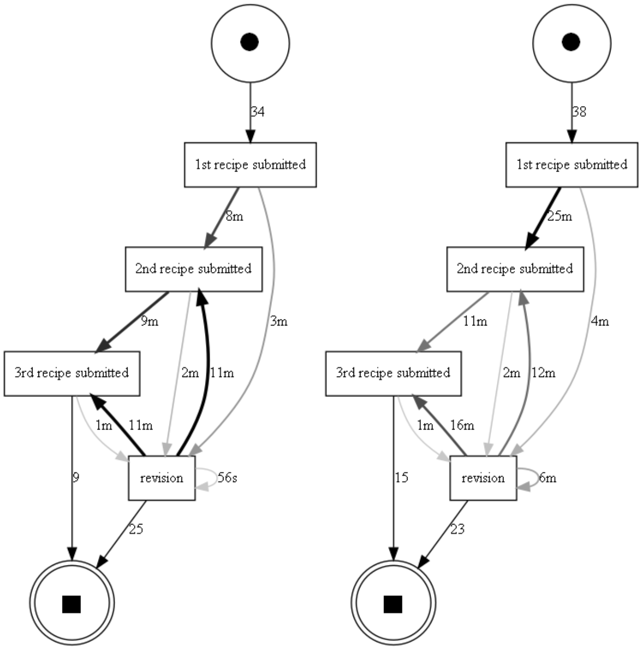


Figure 3: Overview of the SRL behavior of students revising their texts as directly-followed graphs for G1 (left) and G2 (right) automatically calculated and drawn by our pipeline

Engagement

From the second recipe onwards, we find that users revise less often, perform fewer edits, spend less time revising and type faster (Fig. 4). This stems from users being less engaged in the task at hand. In fact, users spend 67% less time revising in G2 (Fig. 4) (from 264 seconds on average for the first recipe to 86.5 seconds for the third one (Tables 4 and 6)) and 64% less in G1 (from 224 seconds to 81.2). In G2, users perform 74% fewer edits between the first and last recipe (Fig. 4). Users in G2 performed on average 222 edits when revising for the first recipe and only 57 for the third one (Tables 4 and 6). The decrease in pause time for the two groups also declines over time (0.822 to 0.553 seconds on average for G1 and 0.646 to 0.525 seconds for G2), even though participants in G2 consistently maintain a smaller average pause time when revising. This is one interpretation of the results and Fig. 4, another one would be to consider users are improving in this task. On average, pause time for G1 decreased by 32.7% and 18.7% for G2 (Fig. 4). Shorter pause times indicate better understanding of the task requirements and better task planning [34]. This is coherent with the participants' reported changes. As seen in Fig. 5, we found that participants from G2 increasingly reported making no changes to their recipes (36% for the third recipe). In contrast, participants in G1 continued reporting making changes based on the received adaptive feedback. Nevertheless, there was also an increase in the participants in G1 that did not edit the recipe, one participant noted *I didn't edit as much this time as I remembered to add them the first time around.*

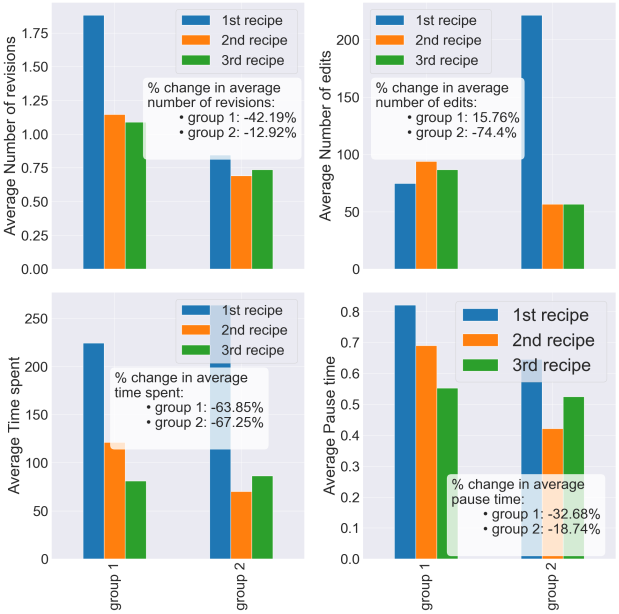


Figure 4: Visualizing user engagement and feature evolution on 4 feature variables over the entirety of the writing experiment

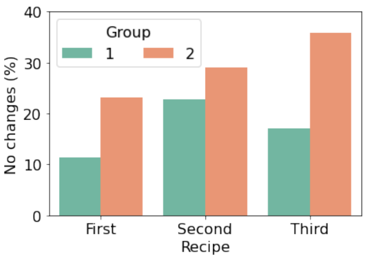


Figure 5: Percentage of participants that stated that they made no changes when editing their recipes in the survey following the experiment

Gender Comparison

Research has often found that males tend to be more impaired at composing text in comparison with women. The study in [35] found that female students performed better than male students on a number of levels. Females had higher scores, revised more, and were more efficient: they revised more per unit of time, exhibiting greater writing fluency. In this study, we found that there is a clear distinction in the writing capabilities between males and females. Like [35], we find females are more efficient in this writing task. They tend to have higher efficiency scores (Fig. 6) and we find $p = 0.0038$ when comparing efficiency scores in G2 (Table 2), which demonstrates the disparity in efficiency distribution between the two groups. Curiously, males revise less often when receiving feedback (as seen on the x-axis by the number of times revised, (Fig. 6)[4]). On the contrary, when users do not receive feedback, females revise once at most (because index 0 is not a revision phase, Fig. 6). This also reinforces women's abilities in their writing, suggesting they feel less need for revision if they do not receive feedback on what they can improve. Regarding the Delete-Insert ratio (DIRatio), although we find there is no statistical difference (Table 2), we find that males in G2 generally have higher scores, especially in G2. Having higher DIRatio scores means users delete a larger portion of their texts (over 15% for several male users in G2, Fig. 6). Looking back at SRL, especially on the critical thinking aspect [6], which is defined as the ability to examine material, we can see males are more self-critical and delete a larger portion of their texts compared to females when they do not receive feedback.

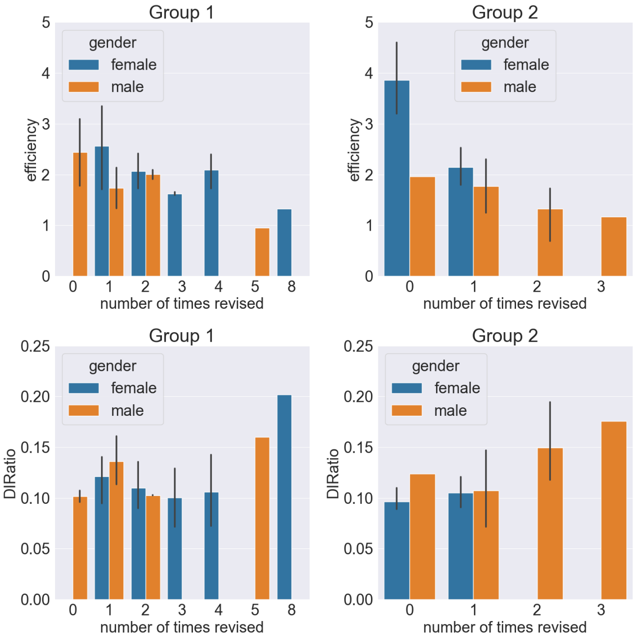


Figure 6: Overview of 4 SRL features from our pipeline comparing males and females

Table 2: p-values for Efficiency and DIRatio features comparing males and females in each group; **p<0.05, **p<0.01, ***p<0.001*

	With Adaptive Feedback (G1)	Without Adaptive Feedback (G2)
Efficiency	$p = 0.215$	$p = 0.0038$**
DIRatio	$p = 0.387$	$p = 0.088$

5. DISCUSSION & CONCLUSION

With this research, we contribute to the field of understanding the use of intelligent writing systems by learners. We do this by gaining insights into their SRL by inspecting revision behavior. From the log data we collected, we built and modelled a pipeline to analyze and visualize user behavior in the revision phases of the writing task, by observing different features extracted from the revisions of G1 (with adaptive feedback) and G2 (without adaptive feedback)(Figs. 2, 4 and 6). Our analysis revealed that learners in different groups revise using different strategies. Learners who were equipped with adaptive feedback revised more often, with fewer edits at each revision step and users without adaptive feedback followed the opposite trend. This suggests that the support provided by the system may influence revision behavior and how it is used. Additionally, we found users seemed to be improving in the writing task as demonstrated by the post-survey and the data, even though they seem to be less engaged from the evolution of the feature variables in Fig. 4. Finally, we concluded females were more efficient than males in this experiment, by having higher efficiency scores. While there has been research on the effectiveness of such systems in improving writing skills, there is a limited understanding of how users revise their writing when using these tools. To evaluate users' SRL, it is crucial to have a better understanding of how they self-regulate, especially in writing activities, in order to provide them with the correct tools to improve their writing skills and understand the underlying writing process [4, 31].

Regarding future directions, one can focus on clustering revision data in order to gain further insights into the revision behavior in a writing task. We have already done this, by identifying eight reported revisions, including *adding more details*, *changing the structure*, *improving the clarity* or *not making any changes*. Nevertheless, we focus on *not making any changes*, but analyzing other revision reports could help shed light on more differences between the groups. As such, clustering could be used for each group to identify the differences between the two groups or between learners in the same group, to see how users revise when receiving feedback or not.

In conclusion, our research on revision behavior in adaptive writing support systems has shed light on how users in different groups approach revision. The development of a pipeline to study this topic has allowed us to collect and analyze data on user writing and revision activity, leading to the discovery of important patterns and trends. Overall, our study has made a significant contribution to the field by providing a deeper understanding of revision behavior.

[4] Some outliers were removed (e.g., users who spent over 10'000 seconds revising or users who have very low efficiency scores (few edits over a long period of time)).

6. REFERENCES

[1] L. K. Allen, M. E. Jacovina, M. Dascalu, R. D. Roscoe, K. M. Kent, A. D. Likens, and D. S. McNamara. Entering the time series space: Uncovering the writing process through keystroke analyses. *International Educational Data Mining Society*, 2016.

[2] A. Augusto, M. Dumas, M. La Rosa, S. Leemans, and S. vanden Broucke. Optimization framework for dfg-based automated process discovery approaches. *Software and Systems Modeling*, 20:1–26, 08 2021.

[3] A. Berti, S. J. van Zelst, and W. M. P. van der Aalst. Process mining for python (pm4py): Bridging the gap between process-and data science. *CoRR*, abs/1905.06169, 2019.

[4] R. A. Bjork, J. Dunlosky, and N. Kornell. Self-regulated learning: Beliefs, techniques, and illusions. *Annual review of psychology*, 64:417–444, 2013.

[5] A. Bogarín, R. Cerezo, and C. Romero. A survey on educational process mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8, 09 2017.

[6] J. Broadbent and W. L. Poon. Self-regulated learning strategies & academic achievement in online higher education learning environments: A systematic review. *The Internet and Higher Education*, 27:1–13, 2015.

[7] X. Chen, L. Breslow, and J. DeBoer. Analyzing productive learning behaviors for students using immediate corrective feedback in a blended learning environment. *Computers Education*, 117:59–74, 2018.

[8] A. Coenen, L. Davis, D. Ippolito, E. Reif, and A. Yuan. Wordcraft: a human-ai collaborative editor for story writing. *CoRR*, abs/2107.07430, 2021.

[9] J. Emig. The composing processes of twelfth graders. 1971.

[10] L. Flower and J. R. Hayes. A cognitive process theory of writing. *College Composition and Communication*, 32(4):365–387, 1981.

[11] J. Fuente, J. Martínez-Vicente, F. H. Santos, P. Sander, S. Fadda, E. Karagiannopoulou, E. Boruchovitch, and D. Kauffman. Corrigendum: Advances on self-regulation models: A new research agenda through the sr vs er behavior theory in different psychology contexts. *Frontiers in Psychology*, 14:1166478, 03 2023.

[12] K. Gero, A. Calderwood, C. Li, and L. Chilton. A design space for writing support tools using a cognitive process model of writing. In *Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2022)*, pages 11–24, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[13] M. Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.

[14] J. R. Hayes. Modeling and remodeling writing. *Written communication*, 29(3):369–388, 2012.

[15] M. Lee, P. Liang, and Q. Yang. CoAuthor: Designing a Human-AI Collaborative Writing Dataset for Exploring Language Model Capabilities. *arXiv e-prints*, page arXiv:2201.06796, Jan. 2022.

[16] M. Leijten and L. Van Waes. Keystroke Logging in Writing Research: Using Inputlog to Analyze and Visualize Writing Processes. *Written Communication*, 30(3):358–392, July 2013. Publisher: SAGE Publications Inc.

[17] D. McCutchen. A capacity theory of writing: Working memory in composition. *Educational psychology review*, 8(3):299–325, 1996.

[18] P. Mejia-Domenzain, M. Marras, C. Giang, and T. Käser. Identifying and comparing multi-dimensional student profiles across flipped classrooms. In *Artificial Intelligence in Education: 23rd International Conference, AIED 2022, Durham, UK, July 27–31, 2022, Proceedings, Part I*, page 90–102, Berlin, Heidelberg, 2022. Springer-Verlag.

[19] M. Nova. Utilizing grammarly in evaluating academic writing: A narrative research on efl students' experience. *Premise: Journal of English Education and Applied Linguistics*, 7(1):80–96, 2018.

[20] V. Padmakumar and H. He. Machine-in-the-loop rewriting for creative image captioning. *CoRR*, abs/2111.04193:573–586, 2021.

[21] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[22] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics, 2019.

[23] S. Sinharay, M. Zhang, and P. Deane. Prediction of Essay Scores From Writing Process and Product Features Using Data Mining Methods. *Applied Measurement in Education*, 32(2):116–137, Apr. 2019. Publisher: Routledge _eprint: https://doi.org/10.1080/08957347.2019.1577245.

[24] C. K. Stallard. An analysis of the writing behavior of good student writers. *Research in the Teaching of English*, 8(2):206–218, 1974.

[25] N. Vandermeulen, M. Leijten, and L. Van Waes. Reporting writing process feedback in the classroom using keystroke logging data to reflect on writing processes. *Journal of Writing Research*, 12(1):109–139, 2020.

[26] T. Wambsganss, A. Janson, and J. M. Leimeister. Enhancing argumentative writing with automated feedback and social comparison nudging. *Computers Education*, 191:104644, 2022.

[27] T. Wambsganss, T. Kueng, M. Söllner, and J. M. Leimeister. Arguetutor: An adaptive dialog-based learning system for argumentation skills. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021.

[28] T. Wambsganss, C. Niklaus, M. Cetto, M. Söllner, S. Handschuh, and J. M. Leimeister. AL: an adaptive learning support system for argumentation skills. In *Proceedings of the 2020 CHI Conference on Human*

Factors in Computing Systems, pages 1–14.

[29] T. Wambsganss, C. Niklaus, M. Söllner, S. Handschuh, and J. M. Leimeister. Supporting cognitive and emotional empathic writing of students. In *The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*, volume abs/2105.14815, 2021.

[30] T. Wambsganß, A. Schmitt, T. Mahnig, A. Ott, N. Ngo, J. Geyer-Klingeberg, J. Nakladal, and J. M. Leimeister. The potential of technology-mediated learning processes: A taxonomy and research agenda for educational process mining. 10 2021.

[31] J. Wong, M. Baars, B. B. de Koning, and F. Paas. Examining the use of prompts to facilitate self-regulated learning in massive open online courses. *Computers in Human Behavior*, 115:106596, 2021.

[32] Z. Yuan. Grammatical error correction in non-native English. Technical Report UCAM-CL-TR-904, University of Cambridge, Computer Laboratory, Mar. 2017.

[33] M. Zhang, J. Hao, C. Li, and P. Deane. Classification of writing patterns using keystroke logs. In L. A. van der Ark, D. M. Bolt, W.-C. Wang, J. A. Douglas, and M. Wiberg, editors, *Quantitative Psychology Research*, pages 299–314, Cham, 2016. Springer International Publishing.

[34] M. Zhang, D. Zou, A. Wu, P. Deane, C. Li, B. Zumbo, and A. Hubley. An investigation of the writing processes in timed task condition using keystrokes. *Understanding and investigating response processes in validation research*, pages 321–339, 2017.

[35] M. Zhu, M. Zhang, and P. Deane. Analysis of Keystroke Sequences in Writing Logs. *ETS Research Report Series*, 2019(1):1–16, 2019. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/ets2.12247.

[36] B. Zimmerman. Becoming a self-regulated learner: An overview. *Theory Into Practice*, 41:64–70, 06 2002.

APPENDIX

A. FEATURES

Table 3 gives a detailed description of the features we study in this paper, extracted in our pipeline.

B. DIRECTLY-FOLLOWS GRAPHS

In process mining, a Directly-Follows Graph is a directed graph that represents the sequence of activities in a process based on event logs. Formally, given a set of activities $\mathcal{A}$, an event log $\mathcal{L}$ is a multiset of traces $t \in \mathcal{L}$, where t is a sequence of activities $t = (a_1, a_2, \ ..., a_n)$, with $a_i \in \mathcal{A}$ $1 \leq i \leq n$ [2]. Given the event log $\mathcal{L}$, the DFG $\mathcal{G}$ is a directed graph such that $\mathcal{G} = (V, E)$. V is the set of activities in $\mathcal{L}$: $V = \{a \in \mathcal{A} \mid \exists t \in \mathcal{L} \land a \in t\}$. E is defined as $E = \{(u, v) \in V \times V \mid \exists t = (a_1, a_2, ..., a_n),\ t \in \mathcal{L}\ \land\ a_i = u\ \land\ a_{i+1}\}$ [2].

C. SEPARATING WRITING SESSIONS

Algorithm 1 describes our implementation of session separation. Each participant wrote a first version of their recipes, then revised it, before starting the next recipes. To focus on the revision sessions, we needed to implement a function which captures the indices in the dataset where participants started their recipes. First, we preprocess the submitted text by removing noisy characters, such as punctuation and return the list of sanitized words. Then we use the GloVe model to convert the words to vectors and return one 50-dimensional vector which is the sum of each word embedding. Then we recursively find the indices of new recipes using cosine distance. However, the algorithm is 91% accurate: this is because sometimes users submitted random strings or the revisions led to the algorithm detecting another recipe. We adjusted the missing indices by hand by looking at the dataset.

Algorithm 1 Separating writing sessions using cosine distance

```
function separateSessions
    $model \leftarrow$ GloVeModel
    function getVector($text$)
        $p \leftarrow$ preProcess($text$)  ▷ splits and sanitizes $text$
        $arr \leftarrow$ Initialize an empty list
        for $word \in p$ do
            add $model[word]$ to $arr$ if $word \in model$
        end for
        return np.sum($arr, axis = 0$)             ▷ uses numpy
    end function

    function computeIndices($startIndex, accumulator$)
        $recipes \leftarrow$ retrieve $recipes$ from the dataset
        $size \leftarrow$ the total number of $recipes$
        if $startIndex \geq size - 1$ then
            return $accumulator$
        end if
        $vec \leftarrow$ getVector($recipes[startIndex]$)
        for $n \leftarrow startIndex$ to $size$ do
            $d \leftarrow 1-$cosineDist($vec$, getVector($recipes[n]$))
            if $d < 0.995$ then
                add $n$ to the $accumulator$
                return computeIndices($n$, $accumulator$)
            end if
        end for
    end function
    return computeIndices(0, empty $accumulator$)
end function
```

D. RESULTS

Detailed Results

Tables 4 to 6 report the mean, standard deviation of different feature variables for both groups, as well as p-values.

Table 3: Overview of feature variables automatically calculated through our pipeline to measure SRL behavior of users in their writing exercises based on keystroke logs

Feature Variables	Description
Number Of Revisions	For each user, we count the amount of times they revise each time they write a recipe (i.e., when they submitted then re-edited their texts). This gives a sense of the effort put into the revision phase of the writing task.
Number of edits	The total number of insertions and deletions during a revision step. Insertions are counted as any characters that are typed including whitespaces, and deletions are counted as the number of times the user presses any of the Backspace or Delete buttons.
Time Spent Revising in seconds	We compute the average time users spend revising for each group, for each recipe. This allows us to compare the two groups and to estimate the effort put in by both groups.
Delete-Insert Ratio (DI-Ratio)	The average deletions over insertions ratio, which approximately captures the extent of editing and revision of any kind [35].
Efficiency	Estimated by the number of insertions per second, which indicates a general writing speed. This feature is arguably an indicator of writing fluency [35].
Pause Time during Revision in seconds	For each user, we collect the inter-key time interval and compute the mean of these intervals. This captures the average lag time between two adjacent keystroke actions [35]. This feature captures the effort and persistence level of users.

Table 4: Overview of SRL features from our pipeline for the first written text between students receiving adaptive feedback (G1) and no feedback (G2) based on our data set

	With Adaptive Feedback (G1)		Without Adaptive Feedback (G2)		
Feature Variables	Mean	Std	Mean	Std	p-values
Number of Revisions	1.882	2.166	0.846	0.735	0.0071
Number of Edits	75.734	95.114	222.73	338.574	0.24
Time Spent Revising (sec)	224.48	237.37	264.01	530.47	0.694
Pause Time in Revision (sec)	0.822	0.225	0.646	0.08	0.339

Table 5: Overview of SRL features from our pipeline for the second written text between students receiving adaptive feedback (G1) and no feedback (G2) based on our data set

	With Adaptive Feedback (G1)		Without Adaptive Feedback (G2)		
Feature Variables	Mean	Std	Mean	Std	p-values
Number of Revisions	1.147	1.033	0.692	0.722	0.033
Number of Edits	95.051	192.92	57.52	61.31	0.26
Time Spent Revising (sec)	121.19	148	70.15	120.9	0.11
Pause Time in Revision (sec)	0.69	0.365	0.421	0.2	0.089

Table 6: Overview of SRL features from our pipeline for the third written text between students receiving adaptive feedback (G1) and no feedback (G2) based on our data set

	With Adaptive Feedback (G1)		Without Adaptive Feedback (G2)		
Feature Variables	Mean	Std	Mean	Std	p-values
Number of Revisions	1.12	1.05	0.737	0.676	0.11
Number of Edits	87.61	270.75	57.7	71.7	0.45
Time Spent Revising (sec)	81.2	115.8	86.5	231.9	0.905
Pause Time in Revision (sec)	0.553	0.212	0.525	0.296	0.85

Towards Automated Assessment of Scientific Explanations in Turkish using Language Transfer

Tanya Nazaretsky
Weizmann Institute of Science
Rehovot, Israel
tanya.nazaretsky@weizmann.ac.il

Hacı Hasan Yolcu
Kafkas University
Kars, Türkiye
hasanyolcu@kafkas.edu.tr

Moriah Ariely
Weizmann Institute of Science
Rehovot, Israel
moriah.ariely@weizmann.ac.il

Giora Alexandron
Weizmann Institute of Science
Rehovot, Israel
giora.alexandron@weizmann.ac.il

ABSTRACT

The paper presents a preliminary study on employing Natural Language Processing (NLP) techniques for automated formative assessment of scientific explanations in Turkish, a morphologically rich language with limited educational resources. The proposed method employs zero and few-shot language transfer techniques for creating Turkish NLP models, obviating the need for extensive collection and annotation of Turkish datasets. The study utilizes multilingual BERT-based pre-trained transformer models. It evaluates the effectiveness of different fine-tuning approaches using an existing annotated dataset in Hebrew. The results indicate that, despite being trained using non-perfectly automated translations from Hebrew responses, the best-performing models demonstrated adequate performance when evaluated on authentic Turkish responses. Thus, this research may provide a useful method for building automated scientific explanations assessment models that are transferred between languages.

1. INTRODUCTION

Constructing scientific explanations is one of the core practices in science. Writing good causal explanations in biology requires students to provide a conceptual framework for the observed phenomenon, identify relevant information, infer the unobservable world, grasp underlying causes, and link the causes logically [19, 9, 12]. Biology teachers use open-ended constructive-response items to elicit students' in-depth understanding of scientific concepts and mechanisms. However, answering such open-ended items is a challenging task. Students often struggle to write answers formulated in their own language [17]. Receiving formative feedback that is just and personalized is crucial in allowing students to relate to the missing or wrong parts of their answers and improve their responses accordingly [21, 24, 2]. Natural Language Processing (NLP) holds much promise for automation of this process[28, 5], especially in English [17, 10, 18, 15, 16]. However, for languages like Turkish, Hebrew, and Arabic, a combination of being morphologically rich (where each input token may consist of several functional units, e.g., multiple suffixes and prefixes added to the original word root), and relatively low resource in the educational domain, makes applications of NLP in such languages particularly challenging [25]. To our knowledge, little research exists in this area [1, 3, 6, 4, 8]. [3] proposed a method for automated formative assessment of scientific explanations in Hebrew based on analytic rubrics. [6] presented the first application of Turkish NLP for automated summative assessment of Physics open-ended questions. In the context of summative assessment of short essays in the Arabic language, which is morphologically rich too, [4] used latent semantic analysis and rhetorical structure theory, and [8] used human and automated translation to English to overcome the shortage in Arabic NLP educational resources. We are unfamiliar with more recent research on NLP-based scoring of open-ended questions in Turkish or Arabic. This work is the first step towards NLP-based tools that can support K-12 science educators in providing formative feedback on scientific writing in Turkish. We propose and evaluate a method for creating Turkish NLP models with no need to collect and annotate large datasets in Turkish while using the corresponding annotated dataset in a different language (e.g., Hebrew). Based on this goal, our research questions are formulated as follows:

- Can our models accurately grade unseen responses in Turkish to an item after being trained on Hebrew responses to several items related to the same biological phenomenon?
- Can fine-tuning using a small number of Turkish responses improve the performance of our models?

2. METHODOLOGY

2.1 The instrument

The instrument consisted of two open-ended items about the effect of Smoking and Anemia on the human ability to exercise. Both items refer to the role of red blood cells (RBC) and Hemoglobin, blood circulation, and energy production in cells on humans' physical activity ability. These topics are

T. Nazaretsky, H. H. Yolcu, M. Ariely, and G. Alexandron. Towards automated assessment of scientific explanations in turkish using language transfer. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 453–457, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115770

part of the Israeli and Turkish high school science curricula. The instrument was constructed in English and Hebrew as part of our previous study [3]. One of the authors manually translated it into Turkish (Table 1).

2.2 Data collection

The research population for this study is high school students in Israel and Türkiye. The research sample included 669 Israeli students (25 schools), 10-12 graders, 70% females, and 84 Turkish students (2 schools), 11-graders, 61% females. The instrument was administered to the students by their teachers, who we contacted through teacher professional communities. The data was collected anonymously using an online Google form, the students were requested to fill in their gender, grade, and school name only. In both languages, several correct responses were written by the teachers. In total, 2007 responses in Hebrew and 174 in Turkish were collected.

2.3 Grading rubric and data annotation

This study used the analytic grading rubric created as part of our previous study [3] and aimed at assisting teachers in formative assessment tasks [2]. Each rubric category represents an essential element in the causal chain, constituting a complete scientific explanation. The original rubric consisted of 11 categories. In this study, we used 7 of them (Table 2), excluding the 4 categories challenging for Turkish students yielding highly unbalanced datasets with only 0 to 5 correct answers. We used the grading obtained in our previous study for the Hebrew responses. The Turkish responses were graded as follows. First, two raters (a biology high school teacher and one of the authors) graded all the answers separately according to the analytic rubric mentioned above. Next, the raters resolved all the conflicts and came to a complete agreement.

2.4 Turkish NLP pipeline

2.4.1 BERT language models

Using a transformer deep-learning architecture has led to the development of *few shot learning* - a method of fine-tuning ML models based on very small amounts of annotated data [26] using state-of-the-art language models pre-trained on enormous amounts of textual data in one or several languages. In this research, we employ the few-shot learning approach for sentence classification using several BERT models: the BERT multi-lingual language model pre-trained on the concatenation of Wikipedia in 104 different languages (DistilmBERT[1]) and the BERT model pre-trained on Turkish language (DistilBERTurk[2]) [22], and the Hebrew AlephBert model[3] [23].

2.4.2 Text preprocessing

All the original Hebrew responses were part of the training set. The pre-processing consisted of several steps. First, the Hebrew responses passed automated spelling corrections (e.g., the critical word ”Hemoglobin” was misspelled in tens of different ways) and replacement of the Hebrew acronyms (e.g., ”RBC” was replaced with ”red blood cells”) with the entire words. Second, the responses were Google-translated automatically into Turkish. Third, we examined the quality of the automated translation. Although the translation was not perfect and, in some cases, was even unsatisfactory (e.g., ”Red blood cells contain Hemoglobin to which oxygen *binds*.” was translated as ”Kırmızı kan hücreleri, *kardeşi* oksijenle bağlantılı hemoglobin içerir.” meaning ”Red blood cells contain hemoglobin, which is associated with its *sister* oxygen.”), we decided to proceed with the translated data as is.

2.4.3 Fine-tuning and text augmentation

Data augmentation is a typical solution to the problem of unbalanced and very small datasets (like our Turkish dataset) by generating new examples for the minority classes. The newly generated examples are supposed to be different from the original ones but carry the same semantic meaning and label as an original text. It is shown by previous research that text augmentation can significantly improve the resulting models' performance [14]. This paper employed two standard paraphrasing augmentation techniques: back translation [29, 27] and using hand-crafted rules (fixed heuristics) [7]. The back translation was done by automatically translating[4] the positive examples for each category into 11 languages[5] and back (Table 4). In addition, the following rules were introduced for paraphrasing. First, we replaced the words with similar meanings (e.g., red blood cells ”kırmızı kan hücresi” is a synonym to ”alyuvar” and ”eritrosit” and can also be replaced by ”hemoglobin” in our context) and chemical acronyms and abbreviations with the words (e.g., CO, O2, and ATP were replaced by ”karbonmonoksit”, ”oksijen” and ”enerji” respectively). Second, we combined each positive example (per category) with several negative examples (e.g., the concatenation of the two responses in Table 3 can create an augmented answer with all positive categories.)

2.5 Experimental setup

To answer the research questions, we performed five experiments. To allow a fair comparison between zero-shot and few-shot models, we divided the Turkish responses dataset into 5 folds and ran each experiment 5 times per each fold and category. Each time 4 out of 5 folds were used as a test set (n = 139). The fifth fold (n = 35) was not used in the case of zero-shot experiments (Exp. 1-3) and was used as a source for fine-tuning (referred to as ”few-shot set” below) using authentic Turkish responses (Exp. 4,5). Below we describe each experiment's settings in more detail.

Exp. 1 **Zero-shot with multilingual DistilmBERT.** Both Hebrew training (n=2007) and Turkish test datasets (n = 139) were used as is, without preprocessing.

Exp. 2 **Zero-shot with Hebrew AlephBERT.** The Hebrew training (n = 2007) was used without preprocessing. The Turkish test set (n=139) was auto-translated into Hebrew.

[1] https://huggingface.co/distilbert-base-multilingual-cased
[2] https://huggingface.co/dbmdz/distilbert-base-turkish-cased
[3] https://github.com/OnlpLab/AlephBERT

[4] by GoogleTranslator from *deep_translator* Python package
[5] English (En), Finnish (Fi), German (De), Greek (Gr), Hebrew (Iw), Italian (It), Japanese (Ja), Persian (Fa), Tatar (Tt), Ukrainian (Uk), Uzbek (Uz)

Table 1: The Instrument in Turkish and English.

Turkish version	English version
Anemia Item	
Kan testinde kırmızı kan hücre miktarının az olduğu kişiler anemi hastası olarak tanımlanır. Bu insanlar halsizlikten ve egzersiz yapmaktaki zorluktan şikâyet ederler. Az miktarda kırmızı kan hücrelerine sahip anemi hastalarının egzersiz yaparken zorluk yaşamalarının nedenini açıklayınız.	A person was found to have low levels of red blood cells in his blood test (anemia). This person complained to his doctor about weakness and difficulty to exercise. Explain how low levels of red blood cells make it difficult for people with anemia to exercise.
Smoking Item	
Sigara dumanı karbon monoksit (CO) gibi birçok zararlı maddeyi içermektedir. Sigara içerken CO salınımı olur. CO hemoglobine bağlanmada oksijenden daha etkindir. Sigara içenlerde yüksek CO seviyesi egzersiz yapmayı zorlaştırmaktadır, bu durumu nasıl açıklarsınız?	The smoke from cigarettes contains several harmful substances, including the gas carbon monoxide (CO). CO is released from cigarettes while smoking and has a stronger tendency than oxygen to bind to Hemoglobin. Explain how high levels of CO make it difficult for smokers to exercise.

Table 2: The categories of the analytic rubric for the Anemia and Smoking Items. The + and − signs represent if the category is relevant to the item. The percent indicated the percentage of the correct answers per category.

		Anemia		Smoking	
	Category Name	Item		Item	
a	Changes in oxygen levels that bind to Hemoglobin/RBC	−	−	+	40%
b	The role of Hemoglobin/RBC in oxygen transportation	+	45%	+	23%
c	Changes in oxygen levels in the body (general)	+	29%	+	24%
d	Changes in oxygen levels in the cells (micro level)	+	10%	+	8%
e	Oxygen is a reactant in energy production	+	7%	+	6%
f	Changes in energy/ATP levels	+	9%	+	9%
g	Using the term energy/ATP	+	23%	+	9%

Exp. 3 **Zero-shot with Turkish DistilBERTTurk.** The Hebrew training (n = 2007) was preprocessed and auto-translated into Turkish (Subsection 2.4.2). The Turkish test set (n = 139) was used as is.

Exp. 4 **Few-shot with Turkish DistilBERTTurk.** The training set consisted of the Hebrew training set as in Exp. 3 combined with the few-shot Turkish set (n = 2007 + 35 = 2042). The Turkish test set (n = 139) was used as is.

Exp. 5 **Few-shot by text augmentation with Turkish DistilBERTTurk.** The training set consisted of the Hebrew training set (n = 2007) as in Exp. 3 combined with the *augmented* by backtranslation and application of augmentation rules (Subsection 2.4.3) few-shot Turkish set. The size of the augmented few-shot set varied from 300 to 400 depending on the number of positive examples per category. The Turkish test set (n = 139) was used as is.

We fine-tuned the pre-trained models end-to-end (including all transformer layers, the pooling layer, and the final dense output layer) with the Adam optimizer (learning rate = 2e-6, learning warmup = 600) over 5 epochs to minimize the binary cross-entropy loss which is consistent with typical BERT fine-tuning for text classification [11].

3. RESULTS AND DISCUSSION

The performance of the Multilingual models (Exp. 1) was unsatisfactory (Table 5). We attribute the failure of multilingual models to generalize to the different subject (S), object (O), or verb (V) order in Turkish and Hebrew. Both Turkish and Hebrew have flexibility in word order. For example, the sentence ”Red blood cells carry oxygen to the cells” can be written in Turkish in several ways depending on the connotation of emphasis on the importance of either the subject, object, or verb. However, the typical order in Turkish is SOV. For example, the authentic answer is Turkish ”Alyuvarlar hücrelere oksijen taşır” when translated into English (preserving the word order) would be ”Red blood cells to the cells oxygen carry” However, the typical order in Hebrew would be SVO, as in English. This typological dis-similarity and zero lexical overlaps between Hebrew and Turkish (which use entirely different scripts) possibly reduce the multilingual model's power of zero-shot language transfer between Hebrew and Turkish[20].

Both zero-shot models based on the automated translation (Exp. 2 and Exp. 3) showed a significant improvement over the multilingual models (Table 5). They performed pretty similarly with a slight advantage towards the AlephBERT-based model (Exp. 2). However, in our context, the critical advantage of using DistilBERTTurk is automated translation in the training stage. After the training is completed, the real assessment systems based on the resulting models can work with authentic student responses in Turkish. The above guided our decision to try improving Exp. 3 models by fine-tuning using authentic Turkish responses.

The straightforward fine-tuning of the DistilBERTTurk models (Exp. 4) using a small number (n = 35) of authentic Turkish examples did not improve most models' performance (Table 5). It even was a minor degradation compared to Exp. 3. The fine-tuning of the DistilBERTTurk models (Exp. 5) using augmentation performed similarly to vanilla DistilBERTTurk (Exp. 3). Yet, there was an improvement (from slight to moderate agreement) for the most problematic category b.

4. CONCLUSIONS AND NEXT STEPS

This paper presents the results of a study on the automatic scoring of scientific explanations in Biology conducted in Turkish using state-of-the-art language transfer methods.

Table 3: Example of student answers to Anemia and Smoking Items in Turkish and English with the corresponding gradings.

Typical Student Answers		a	b	c	d	e	f	g
Anemia Item								
Egzersiz yaparken enerjiye ihtiyaç duyarız ve bu enerjiyi oksijenli solunum yapar. Oksijeni hücrelere alyuvarlar taşır. Eğer hemoglobin az olursa oksijenli solunum az olur ve açığa çıkan enerji azalır.	When we exercise, we need energy and oxygen respiration makes this energy. Red blood cells carry oxygen to the cells. If the hemoglobin is low, aerobic respiration will be less and the energy released will decrease.	−	1	0	0	1	1	1
Smoking Item								
Sigara içenler sigaranın yanması sonucu açığa çıkan CO gazına daha fazla maruz kalır. CO gazı O2'nin yerine hemoglobinlere bağlanır. Hücrelere ihtiyacı olan yeterli O2 gazı taşınamaz. Hücre metabolizmasında aksaklıklar gözlenir. Bu sebeple kaslar daha çabuk yorulur. Çabuk yorulduklardından egzersiz yapmayı zorlaştırır.	Smokers are more exposed to the CO gas released as a result of cigarette combustion. CO gas binds to hemoglobins instead of O2. The cells cannot carry enough O2 gas that they need. Disturbances in cell metabolism are observed. For this reason, the muscles get tired more quickly. They get tired quickly, making it difficult to exercise.	1	1	1	1	0	0	0

Table 4: Responses resulting from back translation of the original student response, excluding the duplicates.

Response to Anemia Item	lang
Alyuvarlar küçük olduğu için yeterli o2yi taşıyamaz hücreler yeterli besin ve o2 alamaz (original)	Tr
Alyuvarları küçük olduğu için yeterince oksijen taşıyamazlar. Hücreler yeterli besin ve O2 alamazlar	Fa
Alyuvarlar küçük oldukları için yeterli O2 taşıyamazlar.Hücreler yeterli besin ve O2 alamazlar.	Uz
Alyuvarları küçük olduğu için yeterince O2 taşıyamazlar. Hücreler yeterli besin ve O2 alamazlar	Iw, En
Alyuvarları küçük olduğu için yeterince O2 taşıyamazlar. Hücreler yeterli besin ve O2 alamıyor	Gr, Tt
Kırmızı kan hücreleri küçük oldukları için yeterli O2 taşıyamazlar ve hücreler yeterli besin ve O2 alamazlar.	De, Fi
Kırmızı kan hücreleri küçük oldukları için yeterli O2 taşıyamazlar.Hücreler yeterli besin ve O2 alamazlar.	It, Uk
Kırmızı kan hücreleri küçük oldukları için yeterli oksijeni taşıyamazlar.Hücreler yeterli besin ve oksijeni alamazlar	Ja

Table 5: The results. Category a is not relevant for Anemia Item, so it was evaluated based on Smoking Item only. Kappa correlation values were interpreted using [13]: poor (< 0.00), slight ($0.00 - 0.20$), fair ($0.21 - 0.40$), moderate ($0.41 - 0.60$), good ($0.61 - 0.80$), and very good ($0.81 - 1$).

	Exp. 1			Exp. 2			Exp. 3			Exp. 4			Exp. 5		
Cat.	Acc	F1	Kappa	Acc	F1	Kappa	Acc	F1	Kappa	Acc	F1	Kappa	Acc	F1	Kappa
a	0.76	0.76	0.48	0.87	0.87	0.71	0.89	0.89	0.76	0.86	0.86	0.71	0.89	0.89	0.78
b	0.71	0.74	0.26	0.76	0.79	0.40	0.74	0.78	0.33	0.76	0.80	0.38	0.78	0.80	0.48
c	0.75	0.75	0.38	0.81	0.82	0.45	0.79	0.79	0.43	0.78	0.78	0.43	0.77	0.77	0.41
d	0.91	0.91	0.45	0.98	0.98	0.86	0.98	0.98	0.86	0.98	0.98	0.85	0.97	0.97	0.83
e	0.93	0.93	0.50	0.99	0.99	0.89	0.99	0.99	0.89	0.98	0.98	0.79	0.98	0.98	0.84
f	0.82	0.79	0.35	0.95	0.95	0.78	0.95	0.95	0.74	0.95	0.95	0.71	0.96	0.96	0.74
g	0.85	0.87	0.30	0.98	0.98	0.92	0.97	0.97	0.87	0.96	0.96	0.86	0.96	0.96	0.84
mean	0.82	0.82	0.39	0.90	0.91	0.72	0.90	0.91	0.70	0.90	0.90	0.68	0.90	0.90	0.70

Our models, trained based on a non-perfectly automated translated Hebrew training dataset, were analyzed on authentic responses written in Turkish. Using back translation for text augmentation, the best-performing models achieved good and very good agreement with human raters in 5 out of 7 and moderate agreement in 2 rubric categories. Notably, these two categories (b and c, see Table 2) were also the hardest to achieve satisfactory performance in the original Hebrew models [3].

The main limitation of this study is the size of the dataset used to evaluate the models. We plan to collect additional data in Turkish to check if the results are robust. Our previous study in Hebrew estimated the number of required responses to achieve the satisfactory performance of the models [3] as $500-900$. Following the successful implementation of the back translation augmentation method in Turkish, we plan to investigate if the back translation can significantly reduce these numbers in original Hebrew models.

In Hebrew, our method is already implemented in PeTeL, a free learning management platform serving about a thousand science teachers in Hebrew and Arabic. We consider the presented results as a proof of concept of our ability to generalize our system to other (even very different, like Turkish) languages using language transfer, with no need to collect additional training data. Our next steps are to extend this study to the Arabic language.

5. ACKNOWLEDGMENTS

TN is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship.

6. REFERENCES

[1] M. Ariely, T. Nazaretsky, and G. Alexandron. First Steps Towards NLP-based Formative Feedback to Improve Scientific Writing in Hebrew. In *Proceedings of the 13th International Conference on Educational Data Mining (EDM 2020)*, pages 565–568, 2020.

[2] M. Ariely, T. Nazaretsky, and G. Alexandron. Personalized Automated Formative Feedback Can Support Students in Generating Causal Explanations in Biology. *The Proceeding of the 16th International Conference of the Learning Sciences (ICLS 2022)*, pages 953–956, 2022.

[3] M. Ariely, T. Nazaretsky, and G. Alexandron. Machine Learning and Hebrew NLP for Automated Assessment of Open-Ended Questions in Biology. *International Journal of Artificial Intelligence in Education*, 33(1):1–34, Mar 2023.

[4] A. M. Azmi, M. F. Al-Jouie, and M. Hussain. AAEE–Automated evaluation of students' essays in Arabic language. *Information Processing & Management*, 56(5):1736–1752, 2019.

[5] B. Beigman Klebanov and N. Madnani. Automated evaluation of writing–50 years and counting. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 7796–7810, 2020.

[6] A. Çınar, E. Ince, M. Gezer, and Ö. Yılmaz. Machine learning algorithm for grading open-ended physics questions in turkish. *Education and information technologies*, 25(5):3821–3844, 2020.

[7] C. Coulombe. Text data augmentation made simple by leveraging nlp cloud apis. *arXiv preprint arXiv:1812.04718*, 2018.

[8] W. H. Gomaa and A. A. Fahmy. Automatic scoring for answers to Arabic test questions. *Computer Speech & Language*, 28(4):833–857, 2014.

[9] T. A. Grotzer and B. B. Basca. How does grasping the underlying causal structures of ecosystems impact students' understanding? *Journal of Biological Education*, 38(1):16–29, 2003.

[10] M. Ha, R. H. Nehm, M. Urban-Lurain, and J. E. Merrill. Applying computerized-scoring models of written biological explanations across courses and colleges: prospects and limitations. *CBE—Life Sciences Education*, 10(4):379–393, 2011.

[11] M. Huggins, S. Alghowinem, S. Jeong, P. Colon-Hernandez, C. Breazeal, and H. W. Park. Practical guidelines for intent recognition: Bert with minimal training data evaluated in real-world hri application. In *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, pages 341–350, 2021.

[12] C. Krist, C. V. Schwarz, and B. J. Reiser. Identifying essential epistemic heuristics for guiding mechanistic reasoning in science learning. *Journal of the Learning Sciences*, 28(2):160–205, 2019.

[13] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[14] B. Li, Y. Hou, and W. Che. Data augmentation approaches in natural language processing: A survey. *AI Open*, 2022.

[15] O. L. Liu, C. Brew, J. Blackmore, L. Gerard, J. Madhok, and M. C. Linn. Automated scoring of constructed-response science items: Prospects and obstacles. *Educational Measurement: Issues and Practice*, 33(2):19–28, 2014.

[16] K. Moharreri, M. Ha, and R. H. Nehm. EvoGrader: an online formative assessment tool for automatically evaluating written evolutionary explanations. *Evolution: Education and Outreach*, 7(1):1–14, 2014.

[17] R. H. Nehm, M. Ha, and E. Mayfield. Transforming biology assessment with machine learning: automated scoring of written evolutionary explanations. *Journal of Science Education and Technology*, 21(1):183–196, 2012.

[18] R. H. Nehm and H. Haertig. Human vs. computer diagnosis of students' natural selection knowledge: testing the efficacy of text analytic software. *Journal of Science Education and Technology*, 21(1):56–73, 2012.

[19] J. F. Osborne and A. Patterson. Scientific argument and explanation: A necessary distinction? *Science Education*, 95(4):627–638, 2011.

[20] T. Pires, E. Schlinger, and D. Garrette. How multilingual is multilingual BERT? *arXiv preprint arXiv:1906.01502*, 2019.

[21] K. Ryoo and M. C. Linn. Designing guidance for interpreting dynamic visualizations: Generating versus reading explanations. *Journal of Research in Science Teaching*, 51(2):147–174, 2014.

[22] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

[23] A. Seker, E. Bandel, D. Bareket, I. Brusilovsky, R. S. Greenfeld, and R. Tsarfaty. AlephBERT: a Pre-trained Language Model to Start Off your Hebrew NLP Application, 2021.

[24] C. Tansomboon, L. F. Gerard, J. M. Vitale, and M. C. Linn. Designing Automated Guidance to Promote Productive Revision of Science Explanations. *International Journal of Artificial Intelligence in Education*, 27(4):729–757, Dec 2017.

[25] R. Tsarfaty, D. Seddah, S. Kübler, and J. Nivre. Parsing morphologically rich languages: Introduction to the special issue. *Computational linguistics*, 39(1):15–22, 2013.

[26] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (csur)*, 53(3):1–34, 2020.

[27] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268, 2020.

[28] X. Zhai, Y. Yin, J. W. Pellegrino, K. C. Haudek, and L. Shi. Applying machine learning in science assessment: a systematic review. *Studies in Science Education*, 56(1):111–151, 2020.

[29] Y. Zhang, T. Ge, and X. Sun. Parallel data augmentation for formality style transfer. *arXiv preprint arXiv:2005.07522*, 2020.

Automated Identification and Validation of the Optimal Number of Knowledge Profiles in Student Response Data

Brad Din
Durham University
Durham, United Kingdom
bradley.p.din@durham.ac.uk

Tanya Nazaretsky
Weizmann Institute of Science
Rehovot, Israel
tanya.nazaretsky@weizmann.ac.il

Yael Feldman–Maggor
Weizmann Institute of Science
Rehovot, Israel
yael.feldman-maggor@weizmann.ac.il

Giora Alexandron
Weizmann Institute of Science
Rehovot, Israel
giora.alexandron@weizmann.ac.il

ABSTRACT

It is well–known that personalized instruction can enhance student learning. AI–based education tools can be used to incorporate blended learning in the science classroom, and have been shown to enhance teachers' ability to prescribe this personalization. We utilise cluster analysis to reveal student knowledge profiles from their response data. However, clustering algorithms typically require the number of clusters as a hyperparameter, yet there is no clear method for choosing the optimal number. Motivated by a practical instance of this foundational problem for a group–based personalization tool, this paper discusses several variations of the gap statistic to identify the optimal number of clusters in student response data. We begin with a simulation study where the ground truth is known to evaluate the quality of the identified methods. We then assess their behaviour on real student data and suggest a stability–based approach to validate our predictions. We identify an empirical threshold for the number of observations required for a prediction to be stable. We found that if a dataset had cluster structure, very small subsamples also showed cluster structure – large datasets were only required to discern the number of clusters accurately. Finally, we discuss how the method enables teachers to tailor their personalization according to their class environment or teaching goals.

Keywords

Clustering; Gap Statistic; Personalized Instruction

1. INTRODUCTION

In recent years, the increased usage of digital learning environments has led to the mass collection of student data [3]. The task of translating these data into tangible insights for understanding and improving student learning remains an active challenge. Blending technology into student learning and providing actionable analytics has massive potential to support teachers in adopting personalized pedagogy [4, 24, 32, 45]. Personalized instruction has been shown to significantly enhance learning outcomes by adapting various attributes of the learning procedure, such as the pace and the contents, to the specific needs of the individual students [6, 56]. The recent development of GrouPer, a learning analytics tool, has assisted teachers in implementing more personalized instruction [39]. The tool was co–designed with teachers and separates students into competency-based knowledge profiles. Whilst participating teachers acknowledged the power of personalization, they suggested that individual tailoring would be impractical in real K–12 classrooms, and that 'group–based personalization' would be a viable compromise between individual adaptation and frontal instruction, whilst also supporting social learning. In addition to competency–based profiling of the students, the teachers also requested semantic information explaining the knowledge profile that each cluster represents; providing this information has been shown to enhance teachers' ability to prescribe personalized learning sequences [39]. GrouPer with its group–based personalization strategy is currently being integrated into the PeTeL (**P**ersonalized **Te**aching and **L**earning) environment[1], allowing teachers to blend digital learning resources into their teaching and provide personalized pedagogy. Over 1000 physics, chemistry, and biology teachers have chosen to make the environment accessible to more than 12,000 students in real classrooms since 2018. In order to perform a sound analysis, GrouPer must first identify *how many* unique knowledge profiles a given activity contains. This is an instance of a fundamental problem – deciding on the number of clusters in a dataset. This is relevant for many applications in education [44, 46], such as discovering knowledge profiles, adaptive learning and student modelling [12, 13, 21, 22, 25, 29, 33, 34, 40, 50]. Despite this vast use, the issue of investigating ways to decide on the number of clusters in student response data was not studied in a systematic manner. This is the focus of the current work, which is motivated, as described above, by an actual EDM application.

B. Din, T. Nazaretsky, Y. Feldman-Maggor, and G. Alexandron. Automated identification and validation of the optimal number of knowledge profiles in student response data. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 458–465, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115744

[1] https://stwww1.weizmann.ac.il/petel/en/home-en/

2. BACKGROUND

In our application, the student responses to each activity are binary. The number of responses for each activity may vary from a few hundred responses to many thousands. The datasets are highly dimensional, where the number of questions ranges between 5 to 30. Combined with the inherent noise in human–based data [40], identifying cluster structure, if it exists, is significantly non–trivial. Unsupervised clustering learns the natural groups in a dataset from the raw data alone [20, 26]. This can be difficult, since there is no rigorous definition of a cluster [19]. Cluster analysis is used in a wide range of applications. Outside of education, it has found usage in image recognition [17], healthcare [30] and finance research [14], amongst many others. There are many algorithms in the literature, such as density–based clustering (e.g. DBSCAN [11]), distribution clustering (e.g. Gaussian mixture modelling [8]) and hierarchical clustering [37]. To avoid placing strict assumptions on our data structure, we choose the simple yet robust k–means algorithm [20, 31, 49].

The k–means algorithm takes a predefined number of clusters as a hyperparameter, k. One can initialise the cluster centroids randomly, or choose them strategically to avoid finding a local minima [54]. Each point is assigned to its nearest centroid; each centroid is then updated by taking the mean of all cluster members. This procedure is repeated until convergence. An alternative framework is the k–modes algorithm [7, 18], which updates the centroids by taking the mode of all members, retaining their binary nature. For our application, since there is no inherent meaning to the centroid, we use the more robust k–means algorithm, which we found to provide more reliable clustering than k–modes.

3. METHODOLOGY

A handful of methods exist in the literature to identify the optimum number of clusters within a dataset, denoted k^*. Classical statistical approaches (e.g. silhouette index [47]) have been used for many decades. X–means works alongside k–means to estimate k^* using information criteria [41]. Cluster prediction and validation methods have also been exploited [9]. Information theoretic approaches [51] and eigenvalue decomposition methods [16] have recently been implemented with success. However, the simple gap statistic has remained a consistent contender, and importantly does not require stringent assumptions to be made on the dataset. We follow the approach from Tibshirani [53], measuring the quality of clustering at each value of k. We use the Euclidean metric as a measure for the distance between two observations. For each cluster, we calculate the total distance between all members:

$$D_r = \sum_{i,i' \in C_r} \|\boldsymbol{x}_i - \boldsymbol{x}_{i'}\|^2 = 2n_r \sum_{i \in C_r} \|\boldsymbol{x}_i - \boldsymbol{\mu}_r\|^2. \qquad (1)$$

We reduce the complexity to $\mathcal{O}(n_r)$ by comparing each point to the cluster centroid, $\boldsymbol{\mu}_r$. Taking the sum over all clusters, we obtain the total within–cluster sum of squares (WSS):

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r. \qquad (2)$$

As we increase the number of clusters, this quantity will monotonically decrease. After the optimal number, since all points are already close to a centroid, the total WSS plateaus, creating a sharp 'kink' at the optimum k. Methods of detecting this bend have been developed [48], but can be subjective, particularly for noisy data. To alleviate this, we utilise the gap statistic [53]; a comparison between the true sample data and its expectation under an appropriate null reference distribution, (W_k^*):

$$\mathrm{Gap}(k) = E\left[\log\left(W_k^*\right)\right] - \log\left(W_k\right). \qquad (3)$$

We obtain $E\left[\log\left(W_k^*\right)\right]$ by taking the average of many binary bootstrapped samples. Finally, k^* is selected by considering adjacent values of the gap plot with the selection criterion:

$$k^* = \min_k \left\{\mathrm{Gap}(k) \geq \mathrm{Gap}(k+1) - s_{k+1}\right\}, \qquad (4)$$

where $s_k = \mathrm{sd}_k\sqrt{1 + 1/B}$ and sd_k is the standard deviation of the bootstrap samples. In our work, we took $B = 280$, but we observed no significant difference with varying B. The gap statistic performs well when clusters are well-separated and uniform, but fails when the dataset becomes noisy. Prior work removed the logarithms in Eq. (3) [36]; we observed no benefit in doing so. Finally, the criterion in Eq. (4) is not robust; even if the plot has a clear optimum, the criterion fails to identify it correctly. We identify two methods to successfully overcome both of these issues: the weighted gap and DD–stopping criterion. The weighted gap approach [57] is identical to Tibshirani's approach, but modifies Eq. (2):

$$W_k^* = \sum_{r=1}^{k} D_r^* = \sum_{r=1}^{k} \frac{1}{2n_r(n_r - 1)} D_r. \qquad (5)$$

This robust quantity D_r^* represents the averaged sum of the pairwise distances between all points in cluster r; this averaging reduces sensitivity to outliers. These statistics are interpreted as a comparison between a dataset and a truly unclustered distribution, which is crucial for identifying datasets with no cluster structure. However, the weighted gap statistic is also prone to overestimate the numbers of clusters, even if there is a clear optimum in the curve. We consider the alternative 'DD–stopping criterion' [57], which compares *adjacent* neighbours in the gap curve:

$$k^* = \max\left\{2\mathrm{Gap}(k) - \mathrm{Gap}(k-1) - \mathrm{Gap}(k+1)\right\}. \qquad (6)$$

We have also used this criterion with the Tibshirani gap statistic. We therefore consider four methods: the gap statistic, the weighted gap statistic, and their DD–stopping criterion variants. Their typical outputs are shown in Fig. 1. We note that the DD–comparisons not only estimates the 'dominant' cluster structure, but also suggests multiple local maxima. The gap statistic can also produce local maxima [53]; we only obtained a single maximum in our applications.

On real student data, we do not know the ground truth. We begin with a simple study on five different structures of binary synthetic data. In all cases, the dataset will be a matrix of dimensions $n_\mathrm{s} \times n_\mathrm{f}$, where n_s is the number of student responses and n_f is the number of items within the activity. In the context of this study, we refer to the items as features of the model. The simplest structure, but perhaps most fundamental, is the case when the data has no inherent clustering (Model N). Here, the data is simply noise: we generate a matrix where each entry is uniformly chosen to be either 0 or 1. Well–defined cluster structure (Model WC) is generated by defining a matrix of correct responses and overlaying blocks

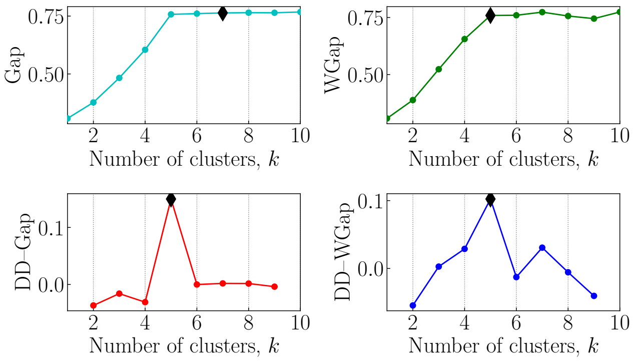


Figure 1: Outputs of the gap statistic, weighted gap statistic, and their DD–variants as a function of the hyperparameter k, shown for synthetic dataset R1 (Table 1).

Table 1: Predicted k^* for a selection of synthetic models. Predictions denoted 'F' failed to satisfy the selection criterion. DD–local maxima are in brackets. Here, $n_s^* = n_s/1000$.

Synthetic Model				k^* Prediction			
Model	n_f	n_s^*	k	Gap	WGap	DD–Gap	DD–WGap
N1	20	1	1	1	1	–	–
WC1	20	1	5	F	8	5	5
WC2	20	1	8	F	F	8	8
UWC1	20	1	5	6	F	5	5
UWC2	5	1	3	3	9	3	3
R1	20	1	5	7	5	5 (3, 5)	5 (5, 7)
R2	5	1	3	F	7	3 (3, 6)	3 (3, 7)
UR1	15	1	3	F	8	3	3 (3, 6, 8)
UR2	15	1	5	F	6	3 (3, 5)	2 (2, 4, 6)
UR3	15	10	5	F	F	5	2 (2, 4, 7)
UR4	15	1	8	F	F	6 (3, 6)	7 (4, 7)
UR5	20	1	5	F	F	5 (5, 8)	2 (2, 5, 7)
UR6	32	1	8	F	F	7 (3, 7)	2 (2, 5, 8)

of incorrect responses along the diagonal. We assume that different clusters have students who are weak in particular skills – a specific block of questions are assumed to measure a particular skill. For k evenly sized clusters, each block has dimensions of $n_s/k \times n_f/k$. To generate psuedo-realistic datasets with noise (Model R), we allow for the probability of students slipping ($P_{\text{slip}} = 0.1$) and guessing ($P_{\text{guess}} = 0.2$) [40]. We generate the background matrix where each entry has a probability of $1 - P_{\text{slip}}$ to be correct. We again overlay incorrect diagonal blocks but allow for the chance of guessing; each entry has a probability of $1 - P_{\text{guess}}$ to being incorrect. Finally, we impose uneven population distributions by defining the k^{th} triangle number, $k_t = k(k+1)/2$. Each cluster population has an increasing fraction of k_t; e.g. cluster n has n/k_t of the total population. This is utilised in the well clustered and realistic synthetic datasets, Models UWC and UR respectively. It is worth noting here that the number of features assigned to each cluster remains constant.

4. RESULTS ON SYNTHETIC DATA

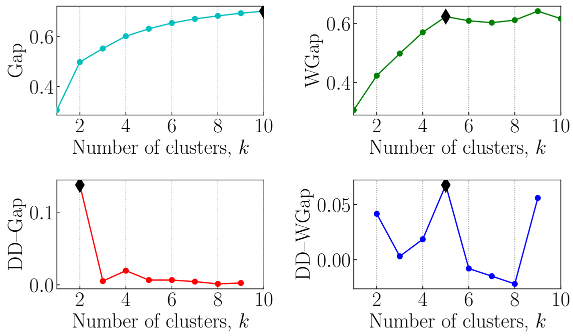


Figure 2: Outputs of the gap statistic, weighted gap statistic, and their DD–variants as a function of the hyperparameter k, shown for the real dataset P2 (Table 4).

A selection of results on synthetic data is shown in Table 1. On unclustered data (Model N), both the gap method and the weighted gap method are able to successfully identify unclustered data. However, on well–clustered data (Model WC), both methods predicted poorly; as can be seen in Fig. 1, the kink of the plot commonly occurs at the correct k, yet the stopping criterion proposed by Tibshirani is unsatisfactory. Both the gap and weighted gap methods were found to suffer from this problem, typically overestimating the number of clusters within the system. The DD–comparison methods were found to solve this issue, performing excellently for data with well–separated and compact clusters. The same results are found with uneven well–clustered data (Model UWC). On more realistic data (Model R), we see similar results. Again, the gap and weighted gap methods are unable to identify the correct number of clusters; however, they were able to identify that some cluster structure exists. The DD–comparison methods again performed well.

Finally, on the uneven realistic data (model UR), we see some interesting results. Model UR1, where each cluster had 5 features, provided the correct prediction. In Models UR2–4, the predicted value from the DD–models was not correct; we can gain some insight by interpreting the 'strength' of a cluster. Models UR2 and UR3 have 3 questions per cluster, whilst Model UR4 has between 1 and 2 questions per cluster. By comparing the labelling of students from the synthetic generation to the labels generated from the clustering, we found that the smallest clusters are prone to being mislabelled and 'absorbed' into the noise of others. Increasing the number of students within this smallest clusters has no effect, as seen in comparing Models UR2 and UR3. We conclude that the strength of a cluster with binary data is determined by the number of questions associated with each cluster – in Models UR5 and UR6, each cluster has 4 features within it and the method is now able to predict correctly. The DD–gap and the DD–weighted gap performed similarly. We therefore adopt a two–step approach: we first apply the gap or weighted gap method to discern if $k > 1$, and then use the DD–comparison method for determining the optimal number of clusters.

5. RESULTS ON STUDENT DATA

The student data considered here was collected from PeTeL activities in a mixture of subjects (Physics, Chemistry) and

Table 2: Predicted k^* for a variety of real student datasets.

Student Dataset			k^* Prediction	
ID Number	n_f	n_s	WGap	DD–WGap
P1	17	1572	4	2 (2, 4)
P2	18	726	5	5 (2, 5, 9)
C1	23	943	4	4 (2, 4, 7, 10)
C2	13	216	4	4 (2, 4, 6, 9)

subtopics (magnetism, forces). An example output on real student data is shown in Fig. 2. Since the signal to noise ratio is now lower, the original gap statistic curve is much shallower. Correspondingly, the DD–Gap method does not provide significantly meaningful predictions, typically finding the optimum number of clusters to be 2; we attribute this to the algorithm identifying the simple splitting of the students into strong/weak groups, which does not represent a meaningful pedagogical contribution. We therefore consider only the weighted gap and DD–weighted gap methods for the remainder of the paper. In Table 4, we show the results on real student datasets, with varying numbers of student responses and items in each learning activity.

Since we do not have a ground truth for these real datasets, we need to assess the validity of these predictions. If we receive a prediction that $k^* > 1$, how do we know that this k^* is correct (true positive)? Conversely, if we receive a prediction that $k^* = 1$, do we require more data (false negative), or does the activity have an inherent unclustered structure (true negative)? Both questions are addressed by considering the *stability* of our prediction. There are many methods of validating the stability of a cluster [9, 27, 52]; we utilise a resampling method used in similar approaches [28]. A stable cluster prediction is one that is similar under a small perturbation to the data (e.g. taking a subsample) [5, 55]. Many methods of cluster stability introduce some figure of merit, typically measuring the similarity between clusterings. We choose a simpler (but more practically–oriented) approach, and compare the predictions of the optimum number of clusters in the resampled dataset. In particular, since we are focusing on the DD–weighted gap method, we consider the predictions for the first 2 local maxima. This has a practical motivation; we do not want to provide teachers with a number of profiles that is too large to manage. We measure the validity of our clustering predictions by repeatedly taking fractional subsamples of our dataset and comparing the prediction results to those of the complete dataset. In order to address the second issue of true/false negatives, since we cannot collect more data, we instead take a dataset which has previously exhibited clustering (e.g. P1) and take subsamples of it. By taking successively smaller fractions, we attempt to identify some quantitative threshold for a 'sufficient' number of student responses.

In Fig. 3, we compare the predictions of the complete dataset to the predictions on three different fractional subsamples of the P1 dataset. Unsurprisingly, the positions of the first two maxima are identical for the largest fraction (90%, corresponding to 1415 students), indicating that the prediction we found was a stable one. We see that there is an increase in variance of the DD–weighted gap plots as we decrease the fraction to 50% (786 students), but the local maxima are again identical. Finally, when we take very small fractions, such as 10% (157 students), we observe significant variance in the DD–weighted gap curve itself, and the position of the local maxima now begin to vary. In Fig. 4, we present the predictions of the first and second local maxima from the DD–weighted gap as a function of the number of students for dataset P1. We infer the stability of each fractional subsample by indicating the frequency of anomalous observations from the complete dataset.

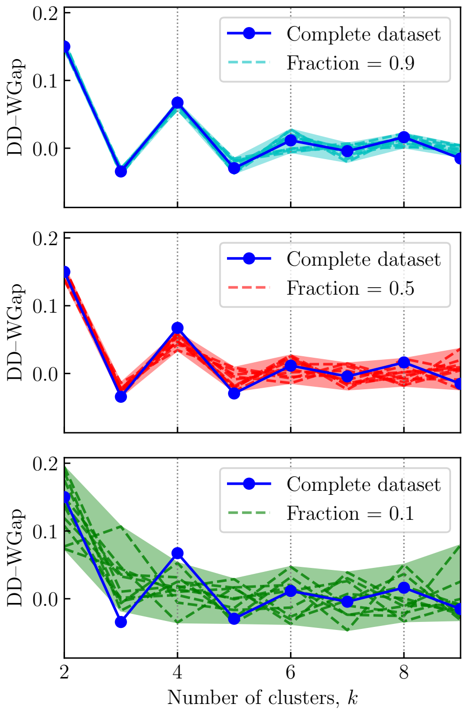


Figure 3: DD–weighted gap plots for three fractions of the P1 dataset: 90% (top), 50% (middle) and 10% (bottom), compared to the full dataset. Each fraction is sampled 10 times.

Our notion of stability allows a prediction on a smaller subsample to be considered stable if the difference is within ±1 of the prediction on the complete dataset, since we expect only a small change after making a small perturbation to the dataset. For the dataset shown in Fig. 4, we find that P1 has a threshold of 550 students. It is worth noting here that similar numerical thresholds were observed in the other clusterable datasets; C2 had a threshold of 660 students, P2 had a threshold of 653, and C3 was found to be unstable immediately. This latter result is not surprising given the small number of observations in the dataset, which is far below the threshold observed in other datasets. Perhaps the most interesting result we found is that the identification of cluster structure required only a remarkably small number of students. Explicitly, when taking 5% of the P1, P2, C2 or C3 datasets (with as few as 30 students), an overwhelming majority of the the weighted gap predictions were still that $k^* > 1$. Although the prediction of k^* in these small fractions was prone to extreme variation, the method was still

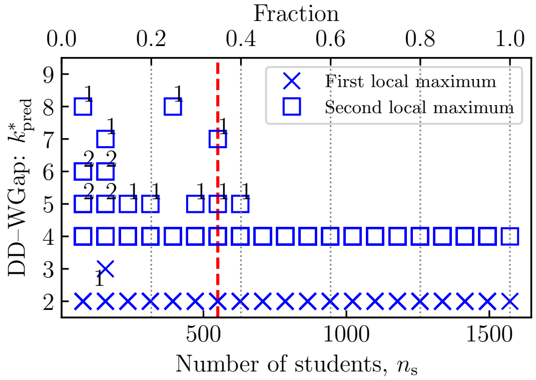


Figure 4: Predictions of the first (crosses) and second (squares) local maxima from the DD–weighted gap method, as a function of number of student observations (sample fraction), for the P1 dataset. If the observation of a fraction was different to that of the complete dataset, then the frequency of each anomalous observation is indicated.

able to confirm that *some* cluster structure existed; very few student responses were needed to discern if a dataset is clusterable. Specifically, it suggests that if an activity (with a reasonable number of responses) is predicted to have $k = 1$, then that particular activity likely *will not* have cluster structure. In this case, one should investigate the specific activity more closely, checking for any issues within the dataset and the data collection procedures itself.

6. DISCUSSION AND CONCLUSIONS

The results on real student data have demonstrated that our approach is able to provide reasonable predictions, proving to be robust even in the presence of noise. We have found that our approach is applicable for a wide range of learning activities. In particular, our stability verification results suggest that the number of responses is not a limiting factor in identifying if cluster structure exists. The method proposed is also completely generic in that it does not rely on any subject–specific knowledge. Although the interpretation of the clusters (e.g. as knowledge profiles) may vary between applications, we expect that this approach should be applicable as a generic tool for identifying cluster structure in a wide range of educational contexts.

A usability-oriented aspect that may influence our decision for the number of clusters is that, in reality, teachers may be constrained in the number of clusters that they are capable of treating simultaneously. This consideration provides a further secondary justification for why the DD–weighted gap method was selected. Providing teachers with multiple good clustering solutions allows them to choose how many clusters they want to work with enables the tool to be useful in a variety of situations; if there are additional teaching assistants in the classroom, or the activities require addtional care and attention, then the teacher may choose to split the class into more/fewer groups as required. Predictions on datasets with an insufficient number of responses will be inaccurate, but may only deviate by a couple of clusters. For our application, it could be argued that it is acceptable to provide teachers with a non-optimal recommendation. Moreoever, the tool is intended to be a recommendation, allowing teachers to override the suggestions if they deem it to be necessary – this is crucial for maintaining trust in the tool [38].

Applying this method in real environments requires careful data collection; it is very easy for a dataset to become very noisy. Some environments allow activities to be customized by teachers, enabling them to remove, modify or rearrange items, inserting inconsistencies into the data. Noise may also result from cheating, making responses unrepresentative of authentic student performance [1, 2]. Such sources of noise (amongst others) are typical for real educational applications [10, 15, 43, 58], and our process handled them in various ways (e.g., excluding activities modified by teachers). We note that the *theoretical* basis for an activity to be suitable for clustering is yet to be established and a better understanding of the types of assessment for which cluster analysis is theoretically justified is an interesting direction for future research. We expect clusters to exist in multi–dimensional activities that involve several binary skills (or skills with very steep learning curve) with some interconnections among them. However, in assessments that make the assumptions of IRT (normally distributed uni/multi–dimensional data), clusters may simply not exist.

In this work, we have evaluated common options for deciding on the optimum number of clusters within a dataset, and discussed their application on binary student data. We have compared these methods on synthetic data where the ground truth is known. We also found some insights into the factors determining the strength of a cluster; the number of features that comprise a cluster is important. This synthetic study formed the basis of our method applied to real student data; we discern if cluster structure exists by using the weighted gap method, and then subsequently determine the precise number of clusters using the DD–weighted gap method, as in [57]. We described an approach to validate the predictions from our method based on fractional resampling [28], and found an empirical threshold for the number of responses to have a stable prediction, typically around 500–600 student observations. Interestingly, we also found that if a data had cluster structure, then the existence of structure was observable with only a small handful of responses. This suggests that large datasets are only important in identifying the precise number of clusters. Our final contribution is the flexibility to the teachers, providing them with options of 'good' clustering solutions that they can apply according to the class environment and pedagogical goals. However, the challenge of providing pedagogically meaningful information about the strengths/weaknesses of each cluster is still outstanding. Methods of providing explanations of the knowledge profiles have already been studied in the literature, automatically building pedagogically meaningful explanations from item-level metadata [23, 35, 42].

7. ACKNOWLEDGEMENTS

BD's work was supported by the Engineering and Physical Sciences Research Council [EP/L01663X/1]. GA's research was supported by The Willner Family Leadership Institute for the Weizmann Institute of Science, Iancovici- Fallmann Memorial Fund, established by Ruth and Henry Yancovich, and by Ullmann Family Foundation. TN is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship.

8. REFERENCES

[1] G. Alexandron, J. A. Ruipérez-Valiente, Z. Chen, P. J. Muñoz-Merino, and D. E. Pritchard. Copying@ Scale: Using harvesting accounts for collecting correct answers in a MOOC. *Computers & Education*, 108:96–114, 2017.

[2] G. Alexandron, L. Y. Yoo, J. A. Ruipérez-Valiente, S. Lee, and D. E. Pritchard. Are MOOC Learning Analytics Results Trustworthy? With Fake Learners, They Might Not Be! *International Journal of Artificial Intelligence in Education*, 29:484506, 2019.

[3] R. Baker and G. Siemens. Educational Data Mining and Learning Analytics. In *The Cambridge Handbook of the Learning Sciences*, pages 253–272. Cambridge University Press, Cambridge, UK, 2014.

[4] R. S. Baker. Stupid tutoring systems, intelligent humans. *International Journal of Artificial Intelligence in Education*, 26(2):600–614, 2016.

[5] A. Ben-Hur and I. Guyon. Detecting Stable Clusters Using Principal Component Analysis. In *Functional Genomics: Methods and Protocols*, pages 159–182. Humana Press, Totowa, NJ, 2003.

[6] B. S. Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational researcher*, 13(6):4–16, 1984.

[7] F. Cao, J. Liang, and L. Bai. A new initialization method for categorical data clustering. *Expert Systems with Applications*, 36(7):10223–10228, 2009.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[9] S. Dudoit and J. Fridlyand. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome biology*, 3(7):1–21, 2002.

[10] A. Dutt, M. A. Ismail, and T. Herawan. A systematic review on educational data mining. *IEEE Access*, 5:15991–16005, 2017.

[11] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press, 1996.

[12] H. Gabbay and A. Cohen. Exploring the Connections Between the Use of an Automated Feedback System and Learning Behavior in a MOOC for Programming. In *Educating for a New Future: Making Sense of Technology-Enhanced Learning Adoption: 17th European Conference on Technology Enhanced Learning, EC-TEL 2022*, pages 116–130, 2022.

[13] H. Gabbay and A. Cohen. Investigating the effect of automated feedback on learning behavior in moocs for programming. In *Proceedings of the 15th International Conference on Educational Data Mining (EDM 2022)*, pages 376–383, 2022.

[14] M. C. Gupta and R. J. Huefner. A cluster analysis study of financial ratios and industry characteristics. *Journal of Accounting Research*, 10:77–95, 1972.

[15] S. Gupta and A. S. Sabitha. Deciphering the attributes of student retention in massive open online courses using data mining techniques. *Education and Information Technologies*, 24(3):1973–1994, 2019.

[16] Z. He, A. Cichocki, S. Xie, and K. Choi. Detecting the Number of Clusters in n-Way Probabilistic Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):2006–2021, 2010.

[17] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy cluster analysis: methods for classification, data analysis and image recognition*. John Wiley & Sons, 1999.

[18] Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, 2(3):283–304, 1998.

[19] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[20] A. K. Jain. Data clustering: 50 years beyond K-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[21] T. Kabudi, I. Pappas, and D. H. Olsen. AI-enabled adaptive learning systems: A systematic mapping of the literature. *Computers and Education: Artificial Intelligence*, 2:100017, 2021.

[22] T. Käser, A. G. Busetto, B. Solenthaler, J. Kohn, M. v. Aster, and M. Gross. Cluster-based prediction of mathematical learning patterns. In *International conference on artificial intelligence in education*, pages 389–399. Springer, 2013.

[23] H. Khosravi, S. B. Shum, G. Chen, C. Conati, Y.-S. Tsai, J. Kay, S. Knight, R. Martinez-Maldonado, S. Sadiq, and D. Gaevi. Explainable Artificial Intelligence in education. *Computers and Education: Artificial Intelligence*, 3:100074, 2022.

[24] J. King and J. South. Reimagining the role of technology in higher education: A supplement to the national education technology plan. *US Department of Education, Office of Educational Technology*, 2017.

[25] S. Klingler, T. Käser, B. Solenthaler, and M. Gross. Temporally coherent clustering of student data. In *Proceedings of the 9th International Conference on Educational Data Mining (EDM 2016)*, pages 102–109, 2016.

[26] S. Križanić. Educational data mining using cluster analysis and decision tree technique: A case study. *International Journal of Engineering Business Management*, 12:1847979020908675, 2020.

[27] T. Lange, V. Roth, M. L. Braun, and J. M. Buhmann. Stability-based validation of clustering solutions. *Neural computation*, 16(6):1299–1323, 2004.

[28] E. Levine and E. Domany. Resampling method for unsupervised estimation of cluster validity. *Neural Computation*, 13(11):2573–2593, 2001.

[29] C. Li and J. Yoo. Modeling student online learning using clustering. In *Proceedings of the 44th Annual Southeast Regional Conference*, ACM-SE 44, page 186191, New York, NY, USA, 2006. Association for Computing Machinery.

[30] M. Liao, Y. Li, F. Kianifard, E. Obi, and S. Arcona. Cluster analysis and its application to healthcare claims data: a study of end-stage renal disease patients who initiated hemodialysis. *BMC nephrology*, 17(1):1–14, 2016.

[31] J. B. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In *Proc. of the*

5th Berkeley Symposium on Mathematical Statistics and Probability, volume 1, pages 281–297. University of California Press, 1967.

[32] R. Martinez-Maldonado, A. Clayphan, K. Yacef, and J. Kay. MTFeedback: Providing Notifications to Enhance Teacher Awareness of Small Group Work in the Classroom. *IEEE Transactions on Learning Technologies*, 8(2):187–200, 2015.

[33] P. Mejia-Domenzain, M. Marras, C. Giang, and T. Käser. Identifying and Comparing Multi-dimensional Student Profiles Across Flipped Classrooms. In *Artificial Intelligence in Education: 23rd International Conference, AIED 2022*, pages 90–102, 2022.

[34] A. Merceron and K. Yacef. Clustering students to help evaluate learning. In *IFIP World Computer Congress, TC 3*, pages 31–42, 2004.

[35] T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

[36] M. Mohajer, K.-H. Englmeier, and V. J. Schmid. A comparison of gap statistic definitions with and without logarithm function, 2011.

[37] F. Murtagh and P. Contreras. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012.

[38] T. Nazaretsky, M. Ariely, M. Cukurova, and G. Alexandron. Teachers' trust in AI-powered educational technology and a professional development program to improve it. *British Journal of Educational Technology*, 53(4):914–931, 2022.

[39] T. Nazaretsky, C. Bar, M. Walter, and G. Alexandron. Empowering Teachers with AI: Co-Designing a Learning Analytics Tool for Personalized Instruction in the Science Classroom. In *LAK22: 12th International Learning Analytics and Knowledge Conference*, pages 1–12, 2022.

[40] T. Nazaretsky, S. Hershkovitz, and G. Alexandron. Kappa learning: A new item-similarity method for clustering educational items from response data. In *Proceedings of the 12th International Conference on Educational Data Mining (EDM 2019)*, pages 129–138, 2019.

[41] D. Pelleg and A. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Icml*, volume 1, pages 727–734, 2000.

[42] M. T. Ribeiro, S. Singh, and C. Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 11351144, New York, NY, USA, 2016. Association for Computing Machinery.

[43] C. Romero, J. R. Romero, and S. Ventura. A survey on pre-processing educational data. In *Educational data mining*, pages 29–64. Springer, 2014.

[44] C. Romero and S. Ventura. Educational data mining: a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(6):601–618, 2010.

[45] C. Romero and S. Ventura. Data mining in education. *WIREs Data Mining and Knowledge Discovery*, 3(1):12–27, 2013.

[46] C. Romero and S. Ventura. Educational data mining and learning analytics: An updated survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(3):e1355, 2020.

[47] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[48] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan. Finding a "Kneedle" in a Haystack: Detecting Knee Points in System Behavior. In *2011 31st International Conference on Distributed Computing Systems Workshops*, pages 166–171, 2011.

[49] D. Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178, 2010.

[50] R. P. Springuel, M. C. Wittmann, and J. R. Thompson. Applying clustering to statistical analysis of student reasoning about two-dimensional kinematics. *Phys. Rev. ST Phys. Educ. Res.*, 3:020107, Dec 2007.

[51] C. A. Sugar and G. M. James. Finding the number of clusters in a dataset. *Journal of the American Statistical Association*, 98(463):750–763, 2003.

[52] R. Tibshirani and G. Walther. Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 14(3):511–528, 2005.

[53] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[54] S. Vassilvitskii and D. Arthur. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, 2006.

[55] U. Von Luxburg et al. Clustering stability: an overview. *Foundations and Trends® in Machine Learning*, 2(3):235–274, 2010.

[56] C. A. Walkington. Using adaptive learning technologies to personalize instruction to student interests: The impact of relevant contexts on performance and learning outcomes. *Journal of educational psychology*, 105(4):932, 2013.

[57] M. Yan and K. Ye. Determining the Number of Clusters Using the Weighted Gap Statistic. *Biometrics*, 63(4):10311037, 2007.

[58] N. Z. Zacharis. A multivariate approach to predicting student outcomes in web-enabled blended learning courses. *The Internet and Higher Education*, 27:44–53, 2015.

APPENDIX

A. ADDITIONAL SYNTHETIC DATA RESULTS

Table 3: Predicted k^* for a selection of synthetic models. Predictions denoted 'F' failed to satisfy the selection criterion. Predictions marked by † were incorrect despite having a clear optimum. DD–local maxima are in brackets. Here, $n_s^* = n_s/1000$.

Synthetic Model				***k^** Prediction**			
Model	n_f	n_s^*	k	Gap	WGap	DD–Gap	DD–WGap
N1	20	1	1	1	1	–	–
N2	20	10	1	1	1	–	–
N3	5	1	1	1	1	–	–
WC1	20	1	5	F†	8†	5	5
WC2	20	1	8	F†	F†	8	8
WC3	8	1	3	F†	3	3	3
WC4	6	1	2	F†	3	2	2
WC5	6	10	2	F†	F†	2	2
UWC1	20	1	5	6†	F†	5	5
UWC2	5	1	3	3	9†	3	3
UWC3	15	1	8	F†	F†	8	8
R1	20	1	5	7	5	5 (3, 5)	5 (5, 7)
R2	5	1	3	F	7	3 (3, 6)	3 (3, 7)
R3	15	1	8	F	F	8 (3, 5, 8)	8 (3, 5, 8)
R4	15	10	8	F	F	8 (3, 5, 8)	8 (3, 5, 8)
UR1	15	1	3	F	8	3	3 (3, 6, 8)
UR2	15	1	5	F	6	3 (3, 5)	2 (2, 4, 6)
UR3	15	10	5	F	F	5	2 (2, 4, 7)
UR4	15	1	8	F	F	6 (3, 6)	7 (4, 7)
UR5	20	1	5	F	F	5 (5, 8)	2 (2, 5, 7)
UR6	32	1	8	F	F	7 (3, 7)	2 (2, 5, 8)

B. ADDITIONAL REAL DATASET RESULTS

Table 4: Predicted k^* for a variety of real student datasets.

Student Dataset			***k^** Prediction**	
ID Number	n_f	n_s	WGap	DD–WGap
P1	17	1572	4	2 (2, 4)
P2	18	726	5	5 (2, 5, 9)
C1	23	943	4	4 (2, 4, 7, 10)
C2	13	216	4	4 (2, 4, 6, 9)
C3	14	292	1	–
C4	14	379	1	–
C5	14	300	1	–
C6	13	241	1	–

The impact of online educational platform on students' motivation and grades: the case of Khan Academy in the under-resourced communities

Ayaz Karimov
Faculty of Information Technology
University of Jyväskylä
akarimov@jyu.fi

Mirka Saarela
Faculty of Information Technology
University of Jyväskylä
mirka.saarela@jyu.fi

Tommi Kärkkäinen
Faculty of Information Technology
University of Jyväskylä
tommi.karkkainen@jyu.fi

ABSTRACT

Even though Azerbaijan is considered a highly educated country from the perspective of schooling years and completed education level, student learning outcomes are underperforming, according to the World Bank. Due to limited resources such as classroom size, access to world-class educational materials, and high-qualified teachers, particularly students from under-resourced communities encounter more challenges during their education life compared to other students who possess these resources. Moreover, online educational platforms play an important role in eliminating learning gaps, particularly in developing countries such as Azerbaijan. In this paper, we describe the implementation and impact of utilizing an online educational platform, the Khan Academy, in one of the under-resourced communities of a developing country. For this, we collaborated with a school in Azerbaijan located in a suburban area. After collecting data through surveys, we applied the association rule mining method. Results from association rule mining concluded that students who studied using the online platform improved their grades and the gamification features of the Khan Academy motivated them. Furthermore, even though it was the first time the school used an online educational platform, almost all students mentioned they would like to learn with these resources in the future. Our study, thus, contributes to how online educational technologies can positively impact the motivation and learning outcomes of students in under-resourced communities.

Keywords

association rule mining, educational technology, gamification, online education, Khan Academy

1. INTRODUCTION

Online learning refers to learning and other supportive resources that are available through a computer. The digital spaces where online learning happens are called online educational platforms (OEPs). OEPs generally contain educational content in different formats such as videos and articles. In some cases, OEPs can also analyze the learning of students based on their interaction inside the platform and provide feedback to improve their learning outcomes [7]. During COVID-19, OEPs played an important role in softening the negative impact of the pandemic on educational activities [2]. The utilization of the OEPs can bring opportunities for students who do not have access to high-quality education, and this can positively alter students' attitudes toward the schools as well as the learning process [6]. After using the OEPs, students enhance their learning performance and become more motivated in the learning process since it guides them to have more meaningful learning behaviors [28]. Students' attitudes towards the OEPs positively change due to various reasons such as being able to track the progress over learning duration, and the possibility of viewing the educational content anytime [4]. One of the reasons why students' attitudes and motivations changed positively is because OEPs started using the gamification elements in their platform to increase engagement [46].

Gamification is the application of game design elements (badges, points, digital coins, etc.) in non-game contexts [15]. [35] and [3] researched the effect of gamification on the motivation of students. They found that gamification motivates students to attempt harder tasks and develop the information literacy skills necessary for success. Moreover, previous studies also found that the utilization of gamification within the learning process can also bring cognitive outcomes. For instance, [30] found that gamification positively affects student retention. They also found that gamification contributes positively to the growth of learners' attitudes and interests at schools. In the research of [19], they designed a gamification plugin to collect students' data and they found that students who completed the assignments in the gamified environments got higher scores. While they made the statement that gamification can possess an emotional and social impact on students that motivate them, gamification may not be the best way to increase motivation for all students. [19] highlighted that gamification environments can also discourage students if they do not acquire the goals within the gamified learning process. In addition to students' motivation and engagement, gamification can also positively impact students' grades [22]. [25] researched the impact of gamification on the students' carefulness. They

A. Karimov, M. Saarela, and T. Kärkkäinen. The impact of online educational platform on students' motivation and grades: the case of khan academy in the under-resourced communities. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 466–473, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115746

found that students indicate a higher level of carefulness when they perform their educational activities in the gamified environment and being more careful towards the assignments increased their grades.

This paper aims to investigate the impact of using OEPs in under-resourced communities and we used Khan Academy as an online educational platform. Khan Academy is one of the largest online educational platforms and it gamifies the learning process by adding gamification elements. Furthermore, our study adds to previous research from different perspectives. Firstly, previous research has been limited to exploring the implementation and impact of OEPs in other countries, and to our knowledge, no prior study has been carried out in Azerbaijan on the topic of the impact of OEPs and their gamification features on students' learning. Secondly, even though the research on the usage of gamification and OEPs in under-resourced communities has been carried out, the number of participants in these studies was limited [1, 21, 50]. However, in our study, 207 students participated within 6 months. Thirdly, recent articles [36, 9, 34] investigated the utilization of Khan Academy and the impact of its gamification features on students' learning. Our study fills the gap by focusing on primary school students in under-resourced communities and we measure how the gamification features impact students' both motivation and grades.

2. LITERATURE REVIEW

2.1 Online educational platforms

Both students and teachers benefit from the OEPs from different perspectives. For teachers, functionalities of OEPs can help in analyzing the learning process and students' learning outcomes in a detailed way [48] and they use OEPs for assigning additional exercises for students who would like to eliminate the learning gaps [33]. For students, OEPs offer the chance to study individually chosen topics and one of the significant advantages is to be able to replay the videos as much as they need which may not be possible at school. Additionally, some OEPs possess a complementary learning experience where the learner can do the follow-up exercises after watching videos or reading articles. Some research showed that using OEPs can positively impact students' learning. [5] mentioned that OEPs improve the students' ability to learn outside the classroom, and if combined effectively, then online and offline learning platforms can help students to understand the subject better. Furthermore, some of the researchers conducted research on the implementation of the OEPs to measure their impact on students' learning. [27] investigated the effect of using online educational content for a month for a math class. They found that there is a positive correlation between the number of studied online educational content and students' achievement. [10] examined the causal impact of online education on the academic performance of students. They found that online educational activities have a positive impact on the exam performance of students. [10] highlight the importance of the content played a crucial role, the lectures that were recorded by higher- quality teachers produced better exam results. In addition to this, [21] and [29] also researched the effect of OEPs on students' performance. Even though they both mentioned the positive impact of OEPs on the students' learning, according to them, other factors should be taken into account within the learning process such as the quality of content and user experience of the platform. Moreover, Khan Academy is also one of these OEPs, and some research concentrated on investigating the utilization and implementation of Khan Academy in the classroom.

2.2 Khan Academy

Khan Academy[1] is a non-profit educational organization created in 2008 by Salman Khan. The organization aims to create a set of online tools that help in providing education to anyone everywhere. Inside the platform, students can watch videos, read articles, and do exercises to study the selected topic. Furthermore, Khan Academy is currently available in more than 50 languages and Azerbaijani is one of them. In our research, we collaborated with the team who leads the localization of the Khan Academy into the Azerbaijani language to measure the impact of the platform on students through a pilot project.

Khan Academy is also one of these OEPs that are used in the classrooms. [9] and [34] measured the impact of using Khan Academy on the grades of students. [9] measured the causal effects of Khan Academy by recruiting 103 students from the 6th and 7th grades. They mentioned that while the expected improvement was 10%, the students showed a 16% improvement in scores. From another perspective, [34] investigated the impact of using Khan Academy on 75 students from the 7th grade and they concluded their research with positive feedback from both students and teachers. According to [34], even though there are better sources to learn, Khan Academy motivates students more since it also includes some engagement features such as badges. Moreover, [23] predicted the effectiveness of Khan Academy's MAP Accelerator which is a mathematics mastery learning platform. They collected data from 181000 students in grades 3-8 across the United States. [23] found that students from high ELL (English Language Learners) districts did not have the same benefits from the use of the MAP Accelerator as their peers. Additionally, according to them, students from these districts were prone to improve 5.3 skills on average per hour, while this number was 7.2 for mid-ELL and 8.9 for low-ELL. Khan Academy also utilizes advanced analytics tools to analyze the learning of students [14]. [42] describes the ALAS-KA provides an extension of the learning analytics support for the Khan Academy platform. ALAS-KA includes also visualized dashboards which allow teachers to analyze the students' learning process. And it also helps students to reflect on their learning.

Gamification is one of the tools that Khan Academy utilizes to increase the learning outcomes of students. Within this framework, they implement various gamification features such as badging, collecting experience points, etc. [36] researched the gamification of computer science content on Khan Academy. According to them, Khan Academy addresses the short-term engagement in the platform successfully by using gamification and this motivates the learner to move further. However, they concluded their research by mentioning the lack of meaningful gamification because this gamification model is not user- centric. [36] mentioned this because in the platform learners collect points without

[1] www.khanacademy.org

matching them to the underlying activities. This does not make the gamification "playful". From another perspective, [41] researched the gamification features of Khan Academy. [41] conducted research on the learning of freshmen students in the topics of physics, chemistry, and mathematics. In this research, they particularly focused on the badging gamification feature of Khan Academy. They found that gaining badges increased their motivation to study more and they felt more motivated by gaining repetition because they were easier to get.

2.3 Gamification

Starting from the early 2010s, gamification was started to be used in the education context to increase learning productivity. Some OEPs utilize such gamification elements to improve the user experience [44]. One example of this can be Duolingo where while learning new words, the user collects the badges and points, then is eventually promoted to the next league in Duolingo [39]. The most common gamification elements are badges, leaderboards, virtual goods, etc. [12], [18], and [17] investigated how one of the gamification elements, badges, impacts students' learning. Badges are graphical symbols or icons given as a reward for certain accomplishments in class such as being an active student, doing all the homework, and getting the badges [20]. According to [12] and [17], using badges positively impact the learning outcomes of students and improves students' grades. They observed that utilizing badges increased the students' motivation in the classroom. Nonetheless, [1] found that utilizing badges can negatively impact the motivation of students. [1] mentioned that when the students earned fewer badges compared to other students, it decreased their motivation.

From another perspective, [38, 11] concentrated on how the leaderboards impact students' motivation and learning productivity. Leaderboards are used to enhance engagement through social comparisons. In the leaderboards, all the participants try to collect points, and based on their points, they are sorted from the most to the least points [31]. Even though the research results of [38], [11] showed leaderboards increase the engagement, motivation, and grades of students, [8] suggests that the implications of leaderboards can also lead to failure. Because, for example, in the case of the research by [32] and [37] leaderboards did not positively impact the motivation of certain groups in the classroom. The reason for this was that students who were at the bottom part of the leaderboard mentioned that it was impossible for them to catch up with the leaders. Thus, at a certain point, they decided to drop out.

Gamification is not limited only to badges, and leaderboards. For instance, [43] used avatars while [13] used virtual goods to increase the motivation of the students. Furthermore, [26] researched the impact of gamification by creating two different groups of students. The first group received a gamified curriculum and the second group received the same curriculum but without gamification elements. Their research resulted with showing the negative impact of gamification on students. They found that students who studied without a gamification-based curriculum scored higher in the final exams. [16] conducted a systematic literature review on the implementation of gamification in education. In this systematic review, [16] found that gamification has a great potential to improve learning if it is designed well and coordinated correctly. They also found that the majority of papers report positive feedback on using gamification in education since it increases the engagement of students, their attendance, and participation in voluntary activities. [24] researched a case study where a student dropped out the school and demonstrated why gamification could change this. [24] discusses that the student was very engaged and motivated about his classes once his teacher was using gamification elements such as rewards. However, with the new academy year, his teacher and their teaching methods changed, then his grades also drastically decreased. [24] hypothesizes that "gamification if conducted globally and interconnected within multiple subjects, can act as a protective factor against early school living."

3. METHODOLOGY

3.1 Context of the study

This research analyzes the impact of using Khan Academy on the learning of students who were part of the pilot project supported by the Ministry of Education of the Republic of Azerbaijan. The project took place in one of the suburban areas of Baku, Azerbaijan, where the graduation level and student participation are lower (around 70%) than in other parts of the country. The project continued from the beginning of October 2021 until the beginning of April 2022. In this pilot project, students were introduced to using the math content of Khan Academy for the primary school levels. The focus group was the 3rd and 4th-grade students. The pilot project was designed in a way that each month around 50 students joined the project. Within this month, they were supposed to learn how to use the platform and then commence studying the topic that they were eager to. Since some students wanted to revise or study the topics that they could not learn in the previous years, they started to study the 1st-grade topic N the platform. Furthermore, at the end of the month, students were offboarded and instead of them, new around 50 students are onboarded. This happened 6 times and in total, 207 participated in the project. Even though Khan Academy holds different gamification features, we collected data about badging and collected experience points of students on the platform. We asked them to highlight three points from their Khan Academy (Appendix C).

3.2 Participants

Students were recruited by sending an information letter to the teachers 15 days in advance. All parents and teachers signed the consent forms to participate in the research. In addition to that, special research permission got also received from the school administration. We provided the tablets to all students and they answered the survey questions with the guidance of teachers. We held an additional online session with teachers to explain to them how the survey should be fulfilled. At last, all the students who participated in the pilot project fulfilled the survey. We gathered data from 207 students who studied in the 3rd or 4th grades (3rd grade students=53.6%, 4th grade students=46.4%). 53.6% of students mentioned that they identify themselves as "male", and the remaining 46.4% selected the "female" option. The vast majority of students (81.2%) participated on all days of the project and only 2.6% of

Table 1: Students' feedback on the future usage of the platform and the platform's impact on their grades

Item	N	Mean	Max value	Min value	Standard deviation
Evaluation of the project	157	4.87	5	2	0.55
Minutes spent on the platform	157	255.06	695	28	189.59
Points collected in the platform	157	26807.62	413839	30	43448.83

students mentioned that they missed the classes more than 5 times. On average, the students spent 255.0641 minutes (SD=189.5876 minutes) on Khan Academy, and the average experience points that the students collected were 43448.83 (SD=26807.62 points).

3.3 Data collection and analysis

We collected the data [2] through surveys. While preparing the questions for the survey, we aimed to collect information about the profiles of students and their performance on Khan Academy (Appendix A). To understand the profiles of students, we asked them to mention their gender, grade, how they evaluate the research project, the number of participation days, whether using Khan Academy changed their grades or not, and their thoughts about using the platform in the future. To collect the responses, we defined several dates (16.04.2022, 18.04.2022, 20.04.2022) with the principal of the school. Because based on the feedback from the principal, the survey was complicated for the students to fulfill by themselves, and they needed the support of teachers. Due to ethical issues, parents had to confirm the participation of the students in the research, and consent was already collected at the beginning of the project when the students joined. Moreover, the main teacher of each class contacted the students to participate in the research. All the students who participated in the pilot project fulfilled the survey in the agreed sessions in the school together with the support of teachers and Khan Academy representatives. To facilitate the process, we also conducted one introductory session for the teachers so that they can answer any upcoming questions from the students. Based on the feedback from the teachers, no problems emerged within the survey fulfilling sessions. Moreover, to analyze the data, we implemented an association rule mining technique where we included variables collected through the survey.

4. RESULTS

4.1 The impact of the platform on the motivation and grades of students

We asked the students to evaluate their experience at Khan Academy. Table 1 demonstrates the responses of students to that question. The students mentioned their thoughts about the platform by giving points from 1 to 5 (1: Very bad, 2: Bad, 3: Normal, 4: Good, 5: Very good). While none of the students mentioned that their experience was very bad, 92.36% of students evaluated Khan Academy as "very good" and the average evaluation score was 4.87. Furthermore, students spent 255.06 minutes and they collected 26807.62 points on the platform on average. Secondly, we asked the students to mention whether they will use Khan Academy in the future. Almost all of the students (96.8%) mentioned that they will use Khan Academy as an additional source to improve their learning outcomes. Subsequently, we measured whether after using Khan Academy, their grades changed. Students could select one of these three options: 1) grade increased; 2) grade remained stable; 3) grade decreased. 68.6% of students answered that their grades increased after using Khan Academy and 29.5% mentioned that their grades did not change.

4.2 Association rules

After the application of the Apriori algorithm, we found three main associations that improved students' motivation and learning. Table 2 indicates the association rules that we found after holding data analysis. The explanation of each variable is explained in Appendix B. The minimum support was 0.5 and the highest support was 0.81 among future_yes and participation_fully variables (confidence=0.98). In Table 2, all rules were generated when the minimum support was 0.5. The confidence in the rules (minimum support=0.5) varied from 51% to 99%. Table 2 also shows the generated association rules, their support, and confidence. From Table 2, we can see that 80.7% of the students, who fully participated in the pilot project, said that they will use Khan Academy in the future. And 67.3% of students, who mentioned that they plan to use Khan Academy in the future, increased their grades. Moreover, we found three major associations that confirmed the positive impact of the pilot project. Firstly, students who earned the Meteorite badge mentioned that they plan to use Khan Academy in the future (support=0.58, confidence=0.97). Meteorite badges are earned in the initial parts of Khan Academy and it is used to motivate the learner. The association that we found shows that earning the Meteorite badge motivated students and they increased their grades. Secondly, the students who fully participated in the classes increased their grades (support=0.62, confidence=0.74). Last but not least, the male students and 3rd- grade students are more prone to utilize Khan Academy in the future (support rate=0.5, confidence=0.98 and support rate=0.53, confidence=0.54 respectively).

When we decreased the minimum support to 0.4, then we also found that students who fully participated in the sessions and increased their grades are more prone to use Khan Academy in the future. Furthermore, the students who received Meteorite badges and fully participated in the classes mentioned that they will utilize the platform in the future. Lastly, based on the generated rules mentioned in Table 2, we can conclude that both full participation in the classes and increasing grades after using Khan Academy motivated students more to use Khan Academy in the future. Moreover, based on the generated association rules, we can see that getting Meteorite badges motivated students to participate in the classes and continue using Khan Academy further. Last but not least, students who identify them-

[2] The datasets generated and analyzed during this study are available from the corresponding author on request.

Table 2: Association rules with support≥ 0.5, their support, and confidence

generated association rules	support	confidence
futureyes → grade3rd	0.53	0.54
futureyes → gendermale	0.5	0.52
participationfully → badgemeteorite	0.5	0.6
participationfully → gradechangegrades_increased	0.62	0.74
participationfully → futureyes	0.81	0.98
futureyes → badgemeteorite	0.58	0.6
futureyes → gradechangegrades_increased	0.67	0.69
participationfully, futureyes → gradechangegrades_increased	0.61	0.75
futureyes, gradechangegrades_increased → participationfully	0.61	0.9
futureyes → participationfully, gradechangegrades_increased	0.61	0.63

selves as ”male” and students from the 3rd grade are more motivated to utilize Khan Academy in the future.

5. DISCUSSION

This paper presents the analysis of implementing online educational platforms to increase the motivation and learning outcomes of students. In this research, we investigated the case of using particularly Khan Academy as a tool to improve students' grades and engagement. [49] and [51] also researched the Khan Academy's impact. [49] mentions that it is very important that the teacher supports the students while using Khan Academy and this research found that Khan Academy can motivate students to do more exercises that directly affect their learning positively. Moreover, [51] highlighted the flipped classroom which included the Khan Academy promoted retention and enhanced students' understanding. In our research, we can also mention that teacher assists students and it brings an extra engagement factor. Students mainly utilized Khan Academy in the school with the guidance and support of the teachers in our research and as [49] mentioned, it helped students not to deal with technical problems rather than focus on the learning process. Furthermore, similar to [49] and [51], we also found that Khan Academy increases the students' motivation. Motivation is one of the most factors to increase learning and while the mentioned authors measured the students' motivation by asking them directly, we analyzed their motivation by asking whether they wanted to use the platform in the future or not [45].

Gamification was also one of the important features that make Khan Academy more engaging from the perspective of students [36]. In our research, we found that students who fully participated in sessions and earned badges (Meteorite badge on Khan Academy) are the ones who also increased their grades. Here, we observe the positive impact of gamification on students' grades. Even though [47] and [40] conducted their research in different regions (Brazil and Spain respectively), they also found that the gamification features of Khan Academy increase the engagement and motivation of students and this directly affects the students' grades.

6. CONCLUSION AND FUTURE WORK

Online educational platforms enhanced students' motivation and learning productivity in different cases. And gamification is also mentioned as one of the most impactful tools to increase engagement inside these online educational platforms. We found that in under-resourced communities, online educational platforms, particularly Khan Academy, positively affect students' grades and motivation. Moreover, gamification increased the willingness of students to spend more time on the platform and use Khan Academy in the future as part of their education. Furthermore, in this research, we focused on primary school students in Azerbaijan, and as an extension of this study, the following points can be investigated. The first potential extension of this paper can be conducting research with secondary school students. Khan Academy helps students to go back to the subjects that students did not understand and study that particular topic. Hereby, secondary school students possess more subjects studied previously. Thus, conducting this research by focusing on secondary school students may indicate the impact of Khan Academy from a different perspective. The second potential extension of this research can be implementing the pilot project within the scope of other social sciences subjects. In our research, math was selected as the subject to implement within the pilot project. Nevertheless, the studying patterns of each subject are various.

Although all the students, who were part of the pilot project that was supported by the Ministry of Education of the Republic of Azerbaijan, fulfilled the survey, these students live in the same community. Thus, the research would provide more detailed results if we were able to collect data from other parts of the country. However, since it was the first time that this project was implemented, only one school was selected. Moreover, due to the scope of the project, we could not collect various information from students. And it affected the minimum support value that we defined. Initially, we defined the support value as 0.7, however, this value did not provide enough association rules to analyze.

6.1 Ethical concerns

Before starting the data collection, we informed both students and parents about the aim of the research and we mentioned that at any phase of the research, they can opt-out to participate and withdraw. Additionally, in the survey, we did not ask any questions that can identify participants. Some parents and students did not want the learners' data to be collected and these students did the same activities with their peers, however, their data were not collected in any form. Moreover, anonymized data were stored in a secure database. Last but not least, there were not any kinds of potential legal, physical, or social harm to students.

7. REFERENCES

[1] S. Abramovich, C. Schunn, and R. M. Higashi. Are badges useful in education?: It depends upon the type of badge and expertise of learner. *Educational Technology Research and Development*, 61(2):217–232, 2013.

[2] O. B. Adedoyin and E. Soykan. Covid-19 pandemic and online learning: the challenges and opportunities. *Interactive Learning Environments*, 31:1–13, 2020.

[3] L. Aguiar-Castillo, L. Hernández-López, P. De Saá-Pérez, and R. Pérez-JIménez. Gamification as a motivation strategy for higher education students in tourism face-to-face learning. *Journal of Hospitality, Leisure, Sport & Tourism Education*, 27:100267, 2020.

[4] N. Ameen, R. Willis, M. N. Abdullah, and M. Shah. Towards the successful integration of e-learning systems in higher education in iraq: A student perspective. *British Journal of Educational Technology*, 50(3):1434–1446, 2019.

[5] W. Bao. Covid-19 and online teaching in higher education: A case study of peking university. *Human behavior and emerging technologies*, 2(2):113–115, 2020.

[6] T. Brahimi and A. Sarirete. Learning outside the classroom through moocs. *Computers in Human Behavior*, 51:604–609, 2015.

[7] S. Carliner. An overview of online learning. 2004.

[8] Y.-k. Chou. *Actionable gamification: Beyond points, badges, and leaderboards*. Packt Publishing Ltd, 2019.

[9] L. Chu, A. Nautiyal, S. Rais, and H. Yamtich. Does khan academy work, 2018.

[10] A. E. Clark, H. Nong, H. Zhu, and R. Zhu. Compensating for academic loss: Online learning and student performance during the covid-19 pandemic. *China Economic Review*, 68:101629, 2021.

[11] M. Ćwil. Leaderboards–a motivational tool in the process of business education. In *Joint International Conference on Serious Games*, pages 193–203. Springer, 2020.

[12] L. da Rocha Seixas, A. S. Gomes, and I. J. de Melo Filho. Effectiveness of gamification in the engagement of students. *Computers in Human Behavior*, 58:48–63, 2016.

[13] L. de Marcos Ortega, A. García-Cabo, and E. G. López. Towards the social gamification of e-learning: A practical experiment. *The International journal of engineering education*, 33(1):66–73, 2017.

[14] Á. Del Blanco, Á. Serrano, M. Freire, I. Martínez-Ortiz, and B. Fernández-Manjón. E-learning standards and learning analytics. can data collection be improved by using standard data models? In *2013 IEEE Global Engineering Education Conference (EDUCON)*, pages 1255–1261. IEEE, 2013.

[15] S. Deterding, D. Dixon, R. Khaled, and L. Nacke. From game design elements to gamefulness: defining" gamification". In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pages 9–15, 2011.

[16] D. Dicheva, C. Dichev, G. Agre, and G. Angelova. Gamification in education: A systematic mapping study. *Journal of educational technology & society*, 18(3):75–88, 2015.

[17] M. Dindar, L. Ren, and H. Järvenoja. An experimental study on the effects of gamified cooperation and competition on english vocabulary learning. *British Journal of Educational Technology*, 52(1):142–159, 2021.

[18] L. Ding. Applying gamifications to asynchronous online discussions: A mixed methods study. *Computers in Human Behavior*, 91:1–11, 2019.

[19] A. Domínguez, J. Saenz-de Navarrete, L. De-Marcos, L. Fernández-Sanz, C. Pagés, and J.-J. Martínez-Herráiz. Gamifying learning experiences: Practical implications and outcomes. *Computers & education*, 63:380–392, 2013.

[20] D. Gibson, N. Ostashewski, K. Flintoff, S. Grant, and E. Knight. Digital badges in education. *Education and Information Technologies*, 20(2):403–410, 2015.

[21] R. Gopal, V. Singh, and A. Aggarwal. Impact of online classes on the satisfaction and performance of students during the pandemic period of covid 19. *Education and Information Technologies*, 26(6):6923–6947, 2021.

[22] S. Grant and B. Betts. Encouraging user behaviour with achievements: an empirical study. In *2013 10th Working Conference on Mining Software Repositories (MSR)*, pages 65–68. IEEE, 2013.

[23] P. Grimaldi, K. Weatherholtz, and K. M. Hill. Estimating the causal effects of Khan Academy Map Accelerator across demographic subgroups. In A. Mitrovic and N. Bosch, editors, *Proceedings of the 15th International Conference on Educational Data Mining*, pages 839–846, Durham, United Kingdom, July 2022. International Educational Data Mining Society.

[24] L. Guerrero-Puerta and M. A. Guerrero. Could gamification be a protective factor regarding early school leaving? a life story. *Sustainability*, 13(5):2569, 2021.

[25] L. Hakulinen, T. Auvinen, and A. Korhonen. The effect of achievement badges on students' behavior: An empirical study in a university-level computer science course. *International Journal of Emerging Technologies in Learning*, 10(1):18–29, 2015.

[26] M. D. Hanus and J. Fox. Assessing the effects of gamification in the classroom: A longitudinal study on intrinsic motivation, social comparison, satisfaction, effort, and academic performance. *Computers & education*, 80:152–161, 2015.

[27] H. J. Heo and M. R. Choi. Flipped learning in the middle school math class. *Advanced Science and Technology Letters*, 71:94–97, 2014.

[28] G.-J. Hwang, S.-Y. Wang, and C.-L. Lai. Effects of a social regulation-based online learning framework on students' learning achievements and behaviors in mathematics. *Computers & Education*, 160:104031, 2021.

[29] S. S. Jaggars and D. Xu. How do online course design features influence student performance? *Computers & Education*, 95:270–284, 2016.

[30] N. F. Jamaludin, T. S. M. T. Wook, S. F. M. Noor, and F. Qamar. Gamification design elements to enhance adolescent motivation in diagnosing depression. *International Journal of Interactive Mobile Technologies*, 15(10):154–172, 2021.

[31] Y. Jia, Y. Liu, X. Yu, and S. Voida. Designing leaderboards for gamification: Perceived differences based on user ranking, application domain, and personality traits. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 1949–1960, 2017.

[32] R. N. Landers, K. N. Bauer, and R. C. Callan. Gamification of task performance with leaderboards: A goal setting experiment. *Computers in Human Behavior*, 71:508–515, 2017.

[33] P. Magalhães, D. Ferreira, J. Cunha, and P. Rosário. Online vs traditional homework: A systematic review on the benefits to students' performance. *Computers & Education*, 152:103869, 2020.

[34] S. Marple, K. Jaquet, A. Laudone, J. Sewell, and K. Liepmann. Khan academy in 7th grade math classes: A case study. *WestEd. org*, 2019.

[35] J. Martí-Parreño, A. Galbis-Córdova, and R. Currás-Pérez. Teachers' beliefs about gamification and competencies development: A concept mapping approach. *Innovations in education and teaching international*, 58(1):84–94, 2021.

[36] B. B. Morrison and B. DiSalvo. Khan academy gamifies computer science. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 39–44, 2014.

[37] M. Ninaus, S. D. Freitas, and K. Kiili. Motivational potential of leaderboards in a team-based math game competition. In *International Conference on Games and Learning Alliance*, pages 242–252. Springer, 2020.

[38] M. Ortiz-Rojas, K. Chiluiza, and M. Valcke. Gamification through leaderboards: An empirical study in engineering education. *Computer Applications in Engineering Education*, 27(4):777–788, 2019.

[39] D. Persico, M. Passarelli, F. Pozzi, J. Earp, F. M. Dagnino, and F. Manganello. Meeting players where they are: Digital games and learning ecologies. *British Journal of Educational Technology*, 50(4):1687–1712, 2019.

[40] J. A. Ruipérez-Valiente, P. J. Muñoz-Merino, C. Delgado Kloos, et al. Detecting and clustering students by their gamification behavior with badges: A case study in engineering education. *International Journal of Engineering Education*, 33(2-B):816–830, 2017.

[41] J. A. Ruipérez-Valiente, P. J. Muñoz-Merino, and C. D. Kloos. Analyzing students' intentionality towards badges within a case study using khan academy. In *Proceedings of the sixth international conference on learning analytics & knowledge*, pages 536–537, 2016.

[42] J. A. Ruipérez-Valiente, P. J. Muñoz-Merino, D. Leony, and C. D. Kloos. Alas-ka: A learning analytics extension for better understanding the learning process in the khan academy platform. *Computers in Human Behavior*, 47:139–148, 2015.

[43] M. Sailer, J. U. Hense, S. K. Mayr, and H. Mandl. How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction. *Computers in human behavior*, 69:371–380, 2017.

[44] M. Sailer and M. Sailer. Gamification of in-class activities in flipped classroom lectures. *British Journal of Educational Technology*, 52(1):75–90, 2021.

[45] U. Schiefele. Interest, learning, and motivation. *Educational psychologist*, 26(3-4):299–323, 1991.

[46] J. Simões, R. D. Redondo, and A. F. Vilas. A social gamification framework for a k-6 learning platform. *Computers in Human Behavior*, 29(2):345–353, 2013.

[47] M. M. Tenório, R. P. Lopes, L. A. d. Góis, and G. d. S. Junior. Influence of gamification on khan academy in brazilian high school. *PUPIL: International Journal of Teaching, Education and Learning*, 2(2):51–65, 2018.

[48] O. Viberg, M. Khalil, and M. Baars. Self-regulated learning and learning analytics in online learning environments: A review of empirical research. In *Proceedings of the tenth international conference on learning analytics & knowledge*, pages 524–533, 2020.

[49] H. E. Vidergor and P. Ben-Amram. Khan academy effectiveness: The case of math secondary students' perceptions. *Computers & Education*, 157:103985, 2020.

[50] B. Yusuf and J. Ahmad. Are we prepared enough? a case study of challenges in online learning in a private higher learning institution during the covid-19 outbreaks. *Advances in Social Sciences Research Journal*, 7(5):205–212, 2020.

[51] Y. Zengin. Investigating the use of the khan academy and mathematics software with a flipped classroom approach in mathematics teaching. *Journal of Educational Technology & Society*, 20(2):89–100, 2017.

APPENDIX

A. SURVEY QUESTIONS

1. What grade are you in? a. 3rd grade b. 4th grade
2. What is your gender? a. male b. female c. other (please specify)
3. How would you evaluate the platform? 1 (Very bad) - 2 (Bad) - 3 (Normal) - 4 (Good) - 5 (Very good)
4. How many days did you participate in the project? a. I participated all days. b. I missed 1-2 days. c. I missed 3-4 days. d. I missed more than 5 days.
5. What badges did you earn on the platform? a. Meteorite b. Moon c. Earth d. Sun e. Black Hole f. Challenge patches
6. How many minutes of study time did you have during the project?
7. How many practice points (XP) did you collect on the platform?
8. How has your math grade changed since using the platform? a. My grades increased. b. My grades decreased. c. My grades remained stable.
9. Do you want to use the platform in the future? a. Yes b. No

B. ACRONYM OF THE VARIABLES

- future_yes: the students who mentioned that they would use Khan Academy in the future
- badge_meteorite: the students who received meteorite badges on Khan Academy

- gradechangegrades_increased: the students who increased their grades during the project
- gendermale: the students who identify themselves as male
- grade3rd: the students who study in the 3rd grade
- participationfully: the students who participated in whole days of the project

C. COLLECTED DATA FROM STUDENTS' PROFILE ON KHAN ACADEMY

- Badges that they earned on the platform. Badges are one of the gamification tools to increase engagement on Khan Academy and active users are awarded badges based on different accomplishments. On Khan Academy, users can earn six various types of badges (Challenge badges - special awards for completing topic challenges on Computing courses; Meteorite badges - common and easy to earn when just getting started; Moon badges - uncommon and represent an investment in learning; Earth badges - require a significant amount of learning; Black Hole badges - the rarest Khan Academy awards; Sun badges - require impressive dedication).
- Experience points (XPs) that they earned. By watching videos, reading articles, and doing exercises, the user can earn points on Khan Academy and we asked the students to highlight how many XPs they earned.
- Learning duration. Khan Academy counts the number of minutes spent on the platform while doing learning activities such as watching videos and solving problems and students mentioned this in the survey.

Measuring Similarity between Manual Course Concepts and ChatGPT-generated Course Concepts

Yo Ehara
Tokyo Gakugei University
ehara@u-gakugei.ac.jp

ABSTRACT

ChatGPT is a state-of-the-art language model that facilitates natural language interaction, enabling users to acquire textual responses to their inquiries. The model's ability to generate answers with a human-like quality has been reported. While the model's natural language responses can be evaluated by human experts in the field through thorough reading, assessing its structured responses, such as lists, can prove challenging even for experts. This study compares an openly accessible, manually validated list of "course concepts," or knowledge concepts taught in courses, to the concept lists generated by ChatGPT. Course concepts assist learners in deciding which courses to take by distinguishing what is taught in courses from what is considered prerequisites. Our experimental results indicate that only 22% to 33% of the concept lists produced by ChatGPT were included in the manually validated list of 4,096 concepts in computer science courses, suggesting that these concept lists require manual adjustments for practical use. Notably, when ChatGPT generates a concept list for non-native English speakers, the overlap increases, whereas the language used for querying the model has a minimal impact. Additionally, we conducted a qualitative analysis of the concepts generated but not present in the manual list.

Keywords

Language Models, Course Concepts, Computer Science

1. INTRODUCTION

ChatGPT is a state-of-the-art natural language processing (NLP)-based artificial intelligence (AI) chatbot system released by OpenAI on November 30, 2022, and can answer any question you enter in a dialogue format. For example, in education, it can be used to answer simple code generation and short essays, and early reports say that the system has surprisingly excellent quality in many tasks. However, its answers may contain factual or logical errors. For codes, essays, and other textual items longer than a sentence, a teacher or expert can read them and find errors. However, for those with simpler structures, such as lists, it is difficult for even teachers to detect errors.

In Massively Open Online Courses (MOOCs), typically, learners can freely choose which courses to take. The concepts taught in MOOCs are important for learners to decide which courses to take because the concepts help learners understand what they should learn in the course and what are prerequisites. Since it is time-consuming for a teacher to create a list of concepts in a course, methods were previously proposed to generate the list directly from course transcripts or course materials [1]. However, even while using these, we still need to collect transcribed courses and materials.

If we ask ChatGPT to "tell us about concepts that will be important in computer science learning," will it be possible to produce a high-quality list of concepts automatically? To determine this, it is necessary to evaluate the quality of ChatGPT's output, but human teachers are not good at evaluating list formats.

2. DATASETS

In this study, we need a list of manually identified concepts. If the concept list is based on use within a specific school or region, it may have been based on assumptions about the educational system of that school or region. For example, a list of concepts from a particular university might include the name of the computer systems of that university, or what is learned in high school in the country where the university resides might be treated as something known by all learners and not included in the list. Since it is undesirable to use such a biased list for evaluation, we used concept lists for MOOCs.

[1] offers an openly available MOOC concept list. Their goal was to create concept lists automatically from course transcripts. For this purpose, the concept lists were manually extracted from course transcripts of eight computer science courses on Coursera, a well-known website for MOOCs in English. The list has 4,096 concepts in total. Subsequent works by [1], such as MOOCCube [2] and MOOCCubeX [3], contain much larger lists of concepts. However, these data are Chinese concepts based on XuetangX, a MOOC system whose courses are predominantly in Chinese. Although English translations of these data sets are also provided, we did not use them in this study because they raise the question of whether the list of concepts used in Chinese courses

Y. Ehara. Measuring similarity between manual course concepts and chatgpt-generated course concepts. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 474–476, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115758

Table 1: Overlap Rate with Manual List.

Lang. for Prompt	Gen. for what students	Overlap rate
English	(not specified)	0.222
English	Japanese	0.315
Japanese	Japanese	0.336

corresponds directly to the list of concepts in English.

Many studies have created academic wordlists or lists of technical terms in English, but it is difficult to strictly define "academic" or technical terms in these studies. Unlike these studies, in this study, we focuse more specifically on course concepts that learners actually learn in online computer lectures. Thus, words such as "introduction," which are academic in the sense that they are often used in academic papers but do not express specific concepts in a field, are excluded from the concepts.

3. EXPERIMENTS

In this study, three lists were created for three use cases, assuming a variety of students. First is the use case in which we want to list the concepts that English-speaking students need to learn when studying computer science in English. Second is the use case in which we want to list the concepts that Non-Native English Speaker (NNS) students need to learn when studying computer science in English. Last is the use case in which we want to do this for NNS students by asking ChatGPT in the students' native language instead of English. Japanese was chosen as the language other than English.

The list was generated using ChatGPT. Input to a language model such as ChatGPT to generate something is called a "prompt". For example, the following prompt was used to ask ChatGPT to list concepts that Japanese students would need in an English computer science course.

- "List 40 concepts that Japanese students need to learn when they study computer science in English online courses on computer science."

The reason for specifying 40 concepts is the length limitation of the answers. However, ChatGPT can also ask questions related to the previous question. Therefore, the following additional prompt will generate a list of 40 concepts that are different from the previous one: "List another set of 40 concepts that differs from the previous one." By entering additional prompts like this, a total of about 120 responses can be obtained for each use case. For English-speaking students, we used the prompt in which the word "Japanese" was simply removed from the aforementioned prompt.

4. RESULTS

Table 1 lists the "overlap rate" as the percentage of concepts generated by ChatGPT included in the list of manually confirmed concepts. Note that the list of manually identified concepts is more comprehensive, since the list of manually identified concepts is 4,096, while only about 120 are generated by ChatGPT. "Lang. used for prompt" indicates the language used for the question, and "Gen. for what students" describes the adjective before the word "student" in the prompt example above, such as "Japanese". As shown in Table 1, the highest percentage was generated for Japanese students in Japanese. Conversely, there was no significant difference in the overlap rate for the languages used, i.e., "Lang. used for prompt".

Table 2: Generated but not in Manual List.

relational database, normalization, bus, decidability, transaction, huffman coding, primitive recursive function, float, array, private key, run-length encoding, captcha, object-oriented programming, turing machine, rest, digital signature, loop, arithmetic coding, brute force

The reason for this is future work. Qualitatively, when the type of student was not specified, the generated concepts tended to have more abbreviations for practical content than for theoretical content. Also, specifying "Japanese students" may have implicitly specified generating concepts for university students because studying abroad is more popular among university students. Table 2 shows the words that were not included in the manually generated list for Japanese students in Japanese. Thus, qualitatively, all words appear to represent "concepts". The reasons why these words were not included are also covered in our future work. Notably, the human-made concept list used in this study was made by annotating words that appeared in the actual spoken lectures. Thus, it could be possible that these concepts, although relevant to the courses, tend to be related but are actually not frequently spoken during courses.

5. DISCUSSION

In this study, the course concept lists generated by ChatGPT were compared to manually generated concept lists. The resulting overlap values between ChatGPT-generated course concepts and manually-created course concepts were low. However, the generated course concept lists do not appear to be low quality since almost all of them represent some concepts of informatics, although the overlap values were low.

Hence, the main result of this study, the overlap values, are limited in its generalizability. The low overlap values could possibly indicate that ChatGPT and other language models cannot generate high-quality course concept lists. However, there are other possibilities, as follows.

First, the generated human course concept list may not be exhaustive, while we employed seemingly the most exhaustive manually-created course concept list to the best of our knowledge. In this case, the overlap values would be low regardless of the performance of ChatGPT in generating course concept lists.

It is also important to note that there is a five-year gap between 2017 when the human-handled course concept list was built [1], and 2022, when ChatGPT was introduced. Hence, it is possible that the low overlap values do not indicate ChatGPT's limited capabilities but rather that the trends in informatics have changed over the past five years.

Furthermore, ChatGPT itself is updated daily. Therefore, if the latest version of ChatGPT is used, it is likely that the overlap values may be improved without any special efforts.

6. CONCLUSIONS

ChatGPT is known for its ability to generate text in a variety of formats. Text fluency can be more easily evaluated by native speakers by reading, while evaluation of list format is difficult for humans. In this study, we evaluated the properties of lecture concept lists, which are important for learners to select lectures, by having ChatGPT generate them. Compared to an exhaustive human list of 4096 lecture concepts in the field of computer science, only up to 33% of the list generated by ChatGPT was included in the human list of lecture concepts. This indicates that the focus of ChatGPT as a lecture concept list is different from the focus of human beings when creating a lecture concept list.

If the number of lecture concept lists is small, there will naturally be lecture concepts that are not included in the list, even if they were created manually. This time, we used the most comprehensive list of lecture concepts in a single field that has been created manually. On the other hand, the list was biased toward one field, computer science. Future work will be to evaluate the generation of lecture concept lists by ChatGPT for other fields as well.

7. ACKNOWLEDGMENTS

This work was supported by JST ACT-X Grant Number JPMJAX2006, Japan. We used the ABCI infrastructure of AIST. We appreciate the anonymous reviewers' valuable comments.

8. REFERENCES

[1] L. Pan, X. Wang, C. Li, J. Li, and J. Tang. Course concept extraction in MOOCs via embedding-based graph propagation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 875–884, Taipei, Taiwan, Nov. 2017. Asian Federation of Natural Language Processing.

[2] J. Yu, G. Luo, T. Xiao, Q. Zhong, Y. Wang, W. Feng, J. Luo, C. Wang, L. Hou, J. Li, Z. Liu, and J. Tang. MOOCCube: A large-scale data repository for NLP applications in MOOCs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3135–3142, Online, July 2020. Association for Computational Linguistics.

[3] J. Yu, Y. Wang, Q. Zhong, G. Luo, Y. Mao, K. Sun, W. Feng, W. Xu, S. Cao, K. Zeng, Z. Yao, L. Hou, Y. Lin, P. Li, J. Zhou, B. Xu, J. Li, J. Tang, and M. Sun. Mooccubex: A large knowledge-centered repository for adaptive learning in moocs. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, CIKM '21, page 4643–4652, New York, NY, USA, 2021. Association for Computing Machinery.

Explainable models for feedback design: An argumentative writing example

Antonette Shibani[*]
University of Technology
Sydney, Sydney, Australia
antonette.shibani@uts.edu.au

Ratnavel Rajalakshmi[†]
Vellore Institute of Technology,
Chennai, India
rajalakshmi.r@vit.ac.in

Srivarshan Selvaraj
Vellore Institute of Technology,
Chennai, India
srivarshan.2019@vitstudent.ac.in

Faerie Mattins
Vellore Institute of Technology,
Chennai, India
faeriemattins.r2019@vitstudent.ac.in

Dhivya Chinnappa
JPMorgan Chase and Co.
dhivya.infant@gmail.com

ABSTRACT

Recent works in educational data mining emphasize the need for producing practical insights that enhance learning. There is particular interest in supporting *student writing* by generating actionable feedback using machine learning algorithms. While algorithmic efficiency is generally sought after in machine learning, it might not be the most important aspect to consider for 'explainability'. This paper presents a predictive model for argumentative writing feedback where explanations supported by Local Interpretable Model-agnostic Explanations (LIME), SHapley Additive exPlanation (SHAP), and logic are derived to generate insights for designing student feedback on argumentative writing. It discusses the computational trade-offs and insights derived that inform writing feedback in practice, with lessons transferable to other contexts.

Keywords

explainable, feedback, predictive models, argumentation, writing, educational data mining, learning analytics, black box

1. INTRODUCTION

A common usage of data in education involves the development of *machine learning models* that can provide predictions, recommendations, and personalised support for learners, connecting fields such as Educational Data Mining (EDM), Artificial Intelligence and EDucation (AIED), and Learning Analytics (LA) [10]. Yet, the complex algorithms in these models create a 'black-box' effect, making the variables that contribute to the final prediction unclear (*Intrinsic opacity*) [2] [6]. This phenomenon is challenged by the emergence of Explainable Artificial Intelligence (XAI) as a field of research for models that offer interpretability and trustworthiness [3] [8].

The need for explainability becomes even more eminent when designing *feedback* for student-facing tools where impact on learning is at the forefront. Feedback-based LA systems generally include the provision of automated feedback to learners that closes the loop from the analytics generated [17] [5]. Automated tools can provide additional feedback to learners in a quick, consistent way at a scale that humans can't provide, although noting that students may engage with it in different ways based on their automated feedback literacy and critical engagement skills [12]. For actionable feedback to be provided by LA tools and to increase learner trust, the foundation lies in explainable LA that can help provide appropriate explanations for the decisions by machine learning models [4].

Argumentation is a critical skill for humans as they routinely engage with conflicting information and inconsistencies arising out of them [1]. Teaching argumentation is often integrated into writing curricula through the use of argumentative essays, with recent efforts in analyzing and providing automated feedback on these essays [15] [16]. While progress has been made in identifying and analyzing argumentation in data sets, for instance using argument mining [7], there is a need for more work on providing actionable feedback to learners to improve their argumentation skills. This can be expanded by the work in *writing analytics* that supports the provision of automated feedback to improve writing skills, where feedback to improve students' higher order competencies such as argumentation has been a recent focus [11].

In this study, we present an approach to designing an explainable machine learning model that supports the provision of feedback to learners in argumentative writing. We discuss the specific case of building a predictive model for argumentative writing quality and explain our approaches and findings examining what works best for explainability and feedback design. We demonstrate exemplary methods for developing explainable models for learner feedback and how it can impact educational practitioners who design this feedback and point out avenues for future work.

[*]Antonette Shibani
[†]Ratnavel Rajalakshmi

A. Shibani, R. Rajalakshmi, S. Selvaraj, F. Mattins, and D. Chinnappa. Explainable models for feedback design: An argumentative writing example. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 477–481, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115750

2. OUR APPROACH

Data for this study came from the Dagstuhl-15512 ArgQuality Corpus [14] - a standard annotated corpus commonly used for argumentation studies. The corpus contained 320 arguments manually coded for 15 dimensions of argumentation quality by three annotators with the overall score metrics: Low, Average, or High. The corpus consisted of 16 different issues (topics for arguments), with a for and against stance for each issue. The data set distribution across the different quality metrics is highly imbalanced, reflecting how this data occurs in the real world. Table 1 shows examples from the dataset.

Our approach to building the prediction model for argumentation quality is as follows. To start with, the arguments were pre-processed by filtering out the non-arguments, removing stop words and punctuation, and stemming the words. The ground truth was established by consolidating the annotations for argumentation quality (low, average, high) only considering rows where at least two annotators agreed on the quality. This process removed inconsistencies in the coding, reducing the number of arguments to 261. The four dimensions identified by authors of the data set as key quality indicators: overall quality, cogency, effectiveness, and reasonableness [14] were taken for modeling as the other sub-dimensions were too fine-grained for automated analysis. The data, vectorized using bag-of-words, was then used to build predictive models for argumentation quality, using two approaches discussed next.

In the baseline approach, the vectorized arguments were used to train Logistic Regression, Decision Tree, Random Forest classifiers, and a Neural Network to predict the overall quality. Hyperparameter tuning was performed using an exhaustive grid search on the Logistic Regression, Decision Tree and Random Forest models, and model parameters used are shown in Table 2. While this approach would likely work for evaluating the overall quality of arguments, it will only be able to provide minimal insight for generating feedback (which is often an end goal when opting for more explainable models).

In the proposed approach, we introduce a *two-stage model* to predict the overall quality of the argument along with the other underlying dimensions (cogency, effectiveness, and reasonableness) to enhance model explainability. Three classifiers (Models 1, 2, and 3) individually trained on the vectorized arguments to predict the three underlying dimensions constituted the first stage of the model. The four machine learning algorithms used in the previous approach were also employed in this context to find the best-performing classifier. The second stage of the model used a single classifier (Model 4) trained on a vector formed by augmenting the one-hot encoding of the underlying dimensions with the vectorized argument to add further context (Training stage 2). This classifier predicts the overall quality. For the final two-stage model, the argument vector was passed to the stage 1 classifiers, and the best-performing models were used for predicting each of the three dimensions (Table 4). These predicted dimensions were encoded and augmented to the original argument text vector, which was then fed to the stage 2 classifier to predict overall quality. The steps are shown in (Figure 1)

We use two existing tools to interpret the models in this study for explainability. The first, Local Interpretable Model-Agnostic Explanations (LIME) offers local explanations by explaining the classifier for a single instance [9]. We used LIME to extract explainable features from the Logistic Regression model predicting argumentation quality in our work. The second, SHapley Additive exPlanation (SHAP) uses Shapely values for finding values of the features that influence the model's scoring. SHAP was used to provide explanations for the Decision Tree model predicting argumentation quality.

3. FINDINGS AND DISCUSSION

A weighted average has been taken for precision, recall, and F_1 score to account for class imbalance to evaluate the results of the baseline model (Table 3). The Decision Tree model, though not the best-performing model, is rule-based and can easily provide explanations for the decisions it makes, hence demonstrated in this study for better explainability. The Bag-of-Words representation was chosen as it provides information on the occurrence of words in the argument and can provide insight into the overall quality of the argument, thus enhancing the explainability of the system. In this model, the decision taken in each node is based on the presence or absence of a particular token in the argument. Using the nodes of the tree one can arrive at a rule-based system to provide feedback to the learner. For instance, a node in the decision tree can indicate as follows: if the argument contains any word containing the token "discov" (discover, discovery, etc.) or "found", then the argument is most likely to be of higher quality. An explanation for these rules might be that the arguments based on discoveries and findings of others are higher quality because they include validated claims. This feedback can then be used to suggest adding evidence or links to supporting research to strengthen the argument made.

The proposed 2-stage model improves the explainability of results using the additional underlying dimensions. The chosen classifiers for each model and their results are displayed in Table 4. Some classifiers like the Logistic Regression classifier were chosen to predict the overall quality as it offers better model explainability. This trade-off for explainability where an easier-to-interpret model is used even if it yielded lower scores than the black box model is a way to tackle the intrinsic opacity in algorithmic decision making [2] [6]. The final two-stage model, after integrating stages one and two, achieves a weighted F_1 score of 0.59. Further exploration of model results can identify insights into words and dimensions that indicate better quality argumentation for improved feedback. This was explored using logistic regression results from the 2-stage model.

The logistic regression model's feature coefficients can reveal the impact of individual words on predicting argument quality. Table 5 shows sample words and their coefficients with the three coefficients corresponding to the three levels of qualities. The word 'found' had the highest coefficient, correlating with average overall quality, suggesting its presence impacted the argument's average quality coding. An argument example with 'found' coded as average is in Table 1, and similar impactful words can be studied for providing feedback. Since the model was trained on the augmented

Table 1: Examples from the dataset with selected rows and columns

id	argument	issue	stance	overall quality
arg219206	Americans spend billions on bottled water every year. Banning their sale would greatly hurt an already struggling economy...	ban-plastic-water-bottles	no-bad-for-the-economy	3 (High)
arg219259	Bottled water is somewhat less likely to be found in developing countries, where public water is least safe to drink...	ban-plastic-water-bottles	no-bad-for-the-economy	2 (Average)
arg219213	Estimates variously place worldwide bottled water sales at between $50 and $100 billion each year, with the market expanding at the startling annual rate of 7 percent...	ban-plastic-water-bottles	yes-emergencies-only	1 (Low)

Table 2: Hyperparameter tuning for baseline models

Model	Parameters
Logistic Regression	'C':1.0, 'dual': False, 'fit_intercept':True, 'penalty':none, 'solver':'sag', 'max_iter':5000
Decision Tree	'criterion': 'gini', 'max_features': 'log2', 'splitter': 'best'
Random Forest	'bootstrap':True, 'class_weight':'balances', 'criterion':gini, 'max_features':none,'n_estimators':300, 'oob_score':False, 'warm_start':False

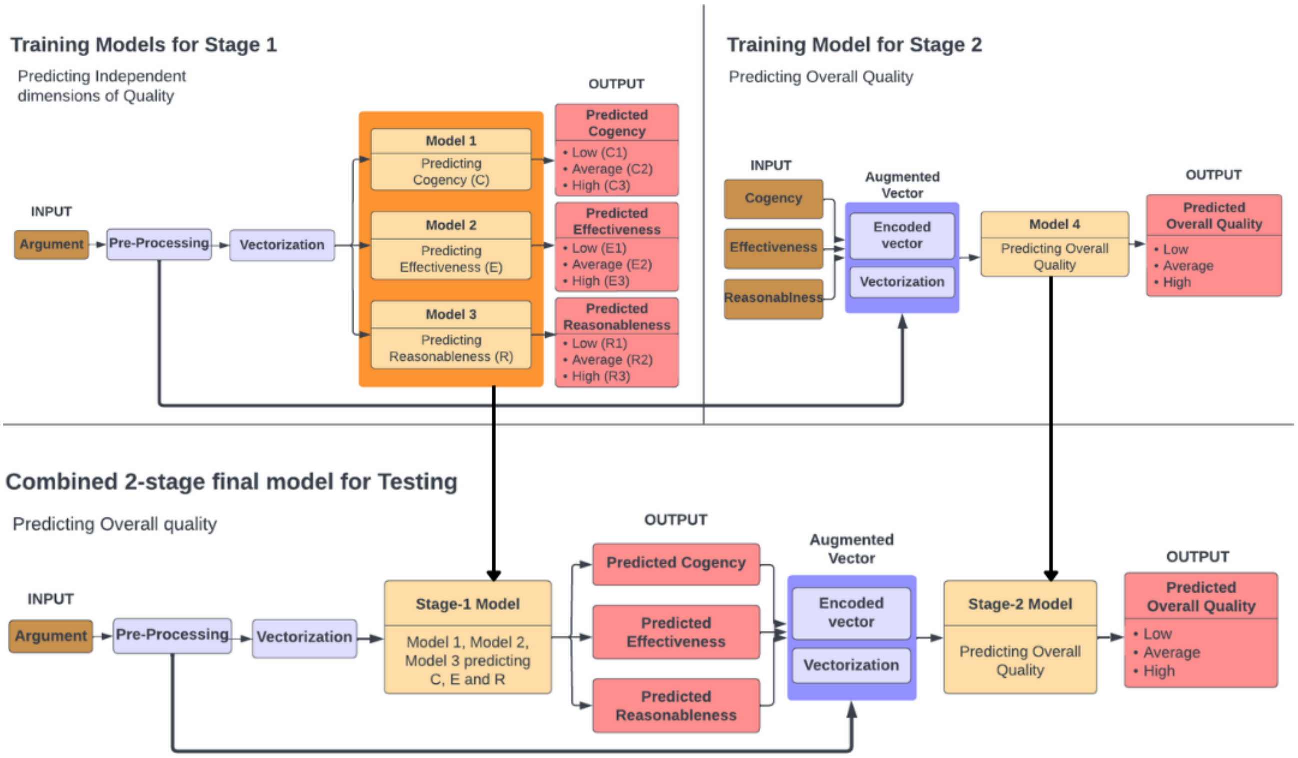


Figure 1: Proposed 2-stage model to predict the overall quality of an argument

vector containing the three underlying dimensions, the same coefficient method can be extended to examine the dimensions as well. From Table 6, we see that if the argument had average effectiveness, then the overall quality of the argument is more likely to be average. Similarly, Reasonableness has the highest positive and negative coefficients, implying its greater impact on overall quality than other dimensions. Thus the feedback provided can be to improve the reasonableness of arguments by explaining the reason behind a stance by using words like ”reason”, ”explain”, and ”because” (derived from the arguments with high reasonableness). Table 6, also displays that Low Cogency contributes the most to Low Overall Quality. Feedback can thus suggest avoiding uncertain language (Words like 'would' and 'think';

Table 3: Performance of the different classifiers in the baseline model for predicting overall quality

Classifier	Accuracy	Precision	Recall	F_1 Score
Logistic Regression	0.62	0.57	0.62	0.59
Decision Tree	0.59	0.58	0.59	0.58
Random Forest	0.62	0.56	0.62	0.58
Neural Network	0.61	0.60	0.61	0.60

Table 4: Performance of the chosen intermediate classifiers

Predicted Dimension	Best Model	Metrics			
		F_1 score	Precision	Recall	Accuracy
Cogency	Neural Network	0.56	0.55	0.58	0.58
Effectiveness	Neural Network	0.56	0.54	0.59	0.59
Reasonableness	Neural Network	0.56	0.55	0.58	0.58
Overall Quality	Logistic Regression	0.87	0.87	0.87	0.87

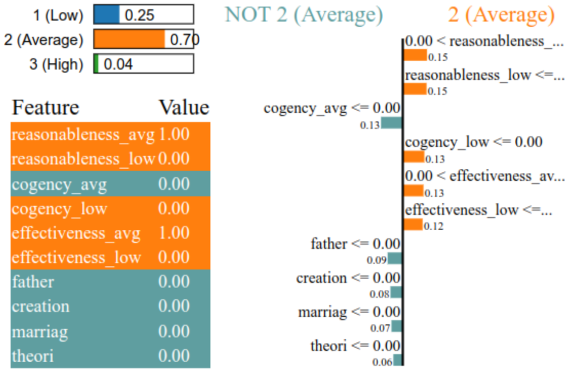
Table 5: Feature coefficients for the word tokens in logistic regression in the 2-stage model

Word	Coef 1 (Low)	Coef 2 (Average)	Coef 3 (High)
discov	-0.024	0.035	-0.011
found	-0.015	0.019	-0.004
although	-0.036	-0.077	0.044

Table 6: Feature coefficients for the underlying dimensions in logistic regression in the 2-stage model.

Dimension	Low	Average	High
Low Cogency	0.335	-0.162	-0.173
Average Cogency	-0.146	0.434	-0.288
High Cogency	-0.189	-0.272	0.461
Low Effectiveness	0.199	-0.090	-0.109
Average Effectiveness	0.045	0.329	-0.374
High Effectiveness	-0.245	-0.239	0.484
Low Reasonableness	0.268	-0.135	-0.132
Average Reasonableness	-0.126	0.513	-0.386
High Reasonableness	-0.141	-0.378	0.519

derived from low cogency arguments) for higher-quality argumentative writing.



Feature	Value
reasonableness_avg	1.00
reasonableness_low	0.00
cogency_avg	0.00
cogency_low	0.00
effectiveness_avg	1.00
effectiveness_low	0.00
father	0.00
creation	0.00
marriag	0.00
theori	0.00

Figure 2: A sample testing instance using LIME for Logistics Regression classifier

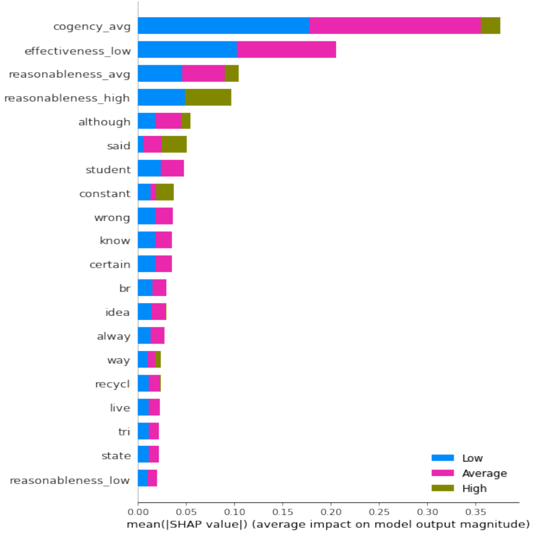


Figure 3: SHAP summary plot for the Decision Tree classifier

Figure 2 demonstrates how LIME can be used to derive explanations for a sample instance, using the 2-stage model to predict the overall quality. The figures display the features and their weights as a table (left) and a bar chart (right), in decreasing order of relevance. The feature 'reasonableness_avg' having a weight of 0.15, is the most significant attribute that supports the instance's average overall quality. The absence of topic-related words (as per the argument's context) such as "father", "creation", "marriag" and "theori" (weights are 0) suggest NOT average overall quality - the presence of such relevant words might indicate higher quality arguments instead. A useful feedback can then be to include more in-depth content related to the topic for higher argumentation quality.

SHAP's summary plot (Figure 3) illustrates the features and their shapely values which attribute more to each target class. The main feature contributing to the prediction of overall argument quality as average is cogency_avg. Similarly, the word 'said' supports the overall quality to be high or average. The word 'idea' contributes to the overall quality being majorly average, possibly pointing to a plan, suggestion, course of action, opinion, or belief, which enhances the argument's overall quality. These frameworks and explanations when evaluated and incorporated into a tool can help generate automated feedback on writing for improving argumentation.

4. CONCLUSION

Our study demonstrates using explainable predictive models for designing feedback for learners. We used a 2-stage model to predict argumentation quality in writing, considering sub-dimensions of quality along with the argument text to enhance explainability. We demonstrated different methods to tackle the intrinsic opacity of algorithms such as the selection of easier-to-interpret models, tailoring the models for particular purposes, choosing features that contribute to better feedback, and decoding model results at different stages to provide actionable feedback. The contribution is hence in presenting an example of a generalisable approach to develop explainable models for feedback. Our methods for using explainable models to inform feedback design apply to various contexts with algorithmic decision-making. These approaches can improve the design of machine learning-based feedback tools that provide learners with interpretable and actionable feedback.

The study is a proof of concept for building explainable models to generate feedback using a small size argumentative writing data set and demonstrated feedback design for the specific context. Future work can build on this work by expanding to larger data sets and examining finer-grained details in the models to provide actionable feedback. While the analysis of the corpus provided insights into argumentation, getting input from educators and co-designing with them is required for a more deliberate design of feedback. This can help validate findings from the model to translate to feedback for classroom practice [13].

5. REFERENCES

[1] P. Besnard and A. Hunter. *Elements of argumentation*, volume 47. MIT press Cambridge, 2008.

[2] J. Burrell. How the machine 'thinks': Understanding opacity in machine learning algorithms. *Big data & society*, 3(1):1–12, 2016.

[3] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang. Xai&#x2014;explainable artificial intelligence. *Science Robotics*, 4(37):eaay7120, 2019.

[4] H. Khosravi, S. B. Shum, G. Chen, C. Conati, Y.-S. Tsai, J. Kay, S. Knight, R. Martinez-Maldonado, S. Sadiq, and D. Gašević. Explainable artificial intelligence in education. *Computers and Education: Artificial Intelligence*, 3:100074, 2022.

[5] K. Kitto, M. Lupton, K. Davis, and Z. Waters. Designing for student-facing learning analytics. *Australasian Journal of Educational Technology*, 33(5):152–168, 2017.

[6] B. Lepri, N. Oliver, E. Letouzé, A. Pentland, and P. Vinck. Fair, transparent, and accountable algorithmic decision-making processes. *Philosophy & Technology*, 31(4):611–627, 2018.

[7] A. Lytos, T. Lagkas, P. Sarigiannidis, and K. Bontcheva. The evolution of argumentation mining: From models to social media and emerging tools. *Information Processing & Management*, 56(6):102055, 2019.

[8] T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[9] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.

[10] B. Rienties, H. Køhler Simonsen, and C. Herodotou. Defining the boundaries between artificial intelligence in education, computer-supported collaborative learning, educational data mining, and learning analytics: A need for coherence. *Frontiers in Education*, 5:128, 2020.

[11] A. Shibani, A. Gibson, S. Knight, P. H. Winne, and D. Litman. Writing analytics for higher-order thinking skills. *Companion Proceedings of the 12th*, page 165, 2022.

[12] A. Shibani, S. Knight, and S. Buckingham Shum. Questioning learning analytics? cultivating critical engagement as student automated feedback literacy. In *LAK22: 12th International Learning Analytics and Knowledge Conference*, pages 326–335, 2022.

[13] A. Shibani, S. Knight, and S. B. Shum. Educator perspectives on learning analytics in classroom practice. *The Internet and Higher Education*, 46:100730, 2020.

[14] H. Wachsmuth, N. Naderi, Y. Hou, Y. Bilu, V. Prabhakaran, T. A. Thijm, G. Hirst, and B. Stein. Dagstuhl-15512-argquality, Apr. 2017.

[15] X. Wang, Y. Lee, and J. Park. Automated evaluation for student argumentative writing: A survey. *arXiv preprint arXiv:2205.04083*, 2022.

[16] W. Xing, H.-S. Lee, and A. Shibani. Identifying patterns in students' scientific argumentation: content analysis through text mining using latent dirichlet allocation. *Educational Technology Research and Development*, 68(5):2185–2214, 2020.

[17] R. Zhi, S. Marwan, Y. Dong, N. Lytle, T. W. Price, and T. Barnes. Toward data-driven example feedback for novice programming. *International Educational Data Mining Society*, 2019.

Fast Dynamic Difficulty Adjustment for Intelligent Tutoring Systems with Small Datasets

Anan Schütt
University of Augsburg
Augsburg, Germany
anan.schuett@informatik.uni-augsburg.de

Tobias Huber
University of Augsburg
Augsburg, Germany
tobias.huber@informatik.uni-augsburg.de

Ilhan Aslan
Huawei Technologies, Munich Research Center
Munich, Germany
ilhan.aslan@huawei.com

Elisabeth André
University of Augsburg
Augsburg, Germany
elisabeth.andre@informatik.uni-augsburg.de

ABSTRACT

This paper studies the problem of automatically adjusting the difficulty level of educational exercises to facilitate learning. Previous work on this topic either relies on large datasets or requires multiple interactions before it adjusts properly. Although this is sufficient for large-scale online courses, there are also scenarios where students are expected to only work through a few trials. In these cases, the adjustment needs to respond to only a few data points. To accommodate this, we propose a novel difficulty adjustment method that requires less data and adapts faster. Our proposed method refits an existing item response theory model to work on smaller datasets by generalizing based on attributes of the exercises. To adapt faster, we additionally introduce a discount value that weakens the influence of past interactions. We evaluate our proposed method on simulations and a user study using an example graph theory lecture. Our results show that our approach indeed succeeds in adjusting to learners quickly.

Keywords

Dynamic difficulty adjustment, Intelligent tutoring system, Computer adaptive practice, Personalized difficulty, Knowledge tracing

1. INTRODUCTION

In computer-based learning, it is important to solidify newly learned content through exercises [16]. Appropriately tailoring the difficulty level of these exercises has a positive effect on learning gains and motivation [8, 23]. Exercises too difficult could lead to anxiety, whereas exercises too easy could lead to boredom, thus the importance of balance, dubbed the state of flow [10]. Consequently, there have been attempts to automatically adjust the difficulty in computer-based learning settings. Individual works refer to this idea using different keywords, such as computer adaptive practice [21, 29], adaptive curriculum [4], or personalized difficulty [41]. In this paper, we use the term Dynamic Difficulty Adjustment (DDA) [18, 26].

Many works on DDA for educational purposes focus on large-scale applications, similar to Massive Open Online Courses (MOOCs). In particular, most of them rely on one of two prerequisites. Either they require large prerecorded datasets to pre-train their models, which can mean up to months' worth of data [4, 21], or they require many interactions per user until they start adapting well [21, 24, 28]. This is not suitable for cases where students only complete a limited number of exercises, for example, when introducing a new concept in higher math or logic. Educational DDA approaches that do not rely on large datasets or many iterations, often break down the learning objective into distinct Knowledge Components (KCs) that students should master [5, 9, 27]. However, defining these KCs can be a laborious task that requires extensive expertise in the subject matter [22]. In certain cases, it is more straightforward to identify exercise attributes instead. For example, in arithmetic exercises, key attributes might include the magnitude of numbers involved or the number of computational steps required. In graph theory, difficulty may hinge on the graph's size and complexity.

For cases where exercise attributes are easier to define than KCs, we propose a novel DDA algorithm based on Item Response Theory (IRT) that alleviates the aforementioned problems of few iterations and small datasets. IRT models are used to predict students' future success in a task based on past interactions [20]. In DDA they can be used to provide a user with exercises that they can solve with a predefined success probability. Traditional IRT models assume that the students have a constant ability level. To fix this, we introduce a discount factor that weakens the influence of past interactions. Because of the lack of massive datasets, the model cannot learn the difficulty of each exer-

A. Schütt, T. Huber, I. Aslan, and E. André. Fast dynamic difficulty adjustment for intelligent tutoring systems with small datasets. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 482–489, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115740

cise individually as IRT-based approaches normally do. By adapting the IRT model to be trained on exercise attributes, our algorithm can generalize the difficulty between exercises. We test our proposed method on an example graph theory lecture with both simulations and a user study. In both experiments, our algorithm succeeds in quickly adjusting the difficulty so that the students obtain our defined success rate.

2. RELATED WORKS

Current DDA methods for exercises in educational settings can be divided into four major techniques. The first category is adapting the difficulty based on handcrafted scoring systems [3, 33, 34]. Here, the students get a score for each completed exercise. This score is then compared to expert-written thresholds to decide which difficulty level will be provided to the student next. The downside of such approaches is that a lot of domain knowledge and time is needed to handcraft good scoring systems for different topics.

The second category is based on the field of Knowledge Tracing (KT). KT addresses the problem of predicting the student's success on an unseen future task given the history of his learning and task attempts [1]. For DDA, KT can be used to predict the success rate of each possible exercise and provide the most suitable one to the student. Leyzberg et al. [24] and Schodde et al. [36], for example, do so by using Bayesian Knowledge Tracing (BKT) which models how likely it is that a student already learned different KCs. Aside from BKT, other models that require KCs to represent exercises include Performance Factor Analysis (PFA) [27] and Additive Factor Model (AFM) [5, 6]. Based on this, BKT, PFA, and AFM can provide exercises for KCs that the student likely has not learned yet. However, if exercises are not distinguishable by different KCs, these models cannot select suitable exercises because all the exercises would be equivalent to the model. In such scenarios, a KT model that can distinguish exercises without relying on KCs is required. The most prominent examples of this are Item Response Theory models (IRT) like the One-Parameter Logistic (1PL) [31] or Four-Parameter Logistic (4PL) [2] model. These models work by learning an ability value for the student and comparing it to the learned difficulty values of each exercise. However, these models are not suitable for DDA since they assume that the student's ability is constant.

The third category of DDA approaches in educational settings is based on the ELO system. ELO was first introduced for chess [14] where it assigns a rating for each player and tries to pit players with similar ratings against each other. After each interaction, the ratings are updated. Klinkenberg et al. [21] were the first to use ELO for DDA. Instead of modeling the ratings of different players, they assign a rating to each individual exercise and each student using the ELO system. In this way, students can be given exercises that match their rating. Recent years saw several variations of ELO-based DDA for education [28, 35, 40]. However, for the scenario we envision, there are two main drawbacks. First, these ELO-based systems require datasets with several repetitions of each individual exercise to learn their individual difficulty rating, with [28] requiring 100 interactions per exercise item. Second, the learning rate in the ELO system is scenario-dependent. Handcrafting such a value is difficult when the goal is fast adaptation without overshooting.

The final category of DDA approaches for education use Reinforcement Learning (RL). Belfer et al. [4] and Zhang et al. [41] use RL to directly choose individual actions that should be provided to the student next. However, both of their approaches require extensive datasets to pre-train their models. Clement et al. [7] used RL to decide whether there should be another more difficult exercise for the same KC or an exercise for another KC. This requires experts to design a carefully crafted curriculum with multiple paths that covers all possible scenarios, which is not feasible for many applications. Another drawback of RL-based DDA approaches is that they require long sequences of interactions to be able to explore different state-action pairs before they can adapt to the student. This makes them unsuitable for scenarios with a limited number of interactions.

In addition to education, there is also a large amount of work on DDA in games. Because of the quick pace of games, many DDA techniques in games can afford to use large amounts of interactions and data [39].For example, Moon et al. [25] used 60.2M data points for pre-training. However, there also have been works on DDA for games that focus on fast adaptation using little data. One group of work here used procedural content generation [11, 12, 17]. These methods rely, at least in part, on the ability to procedurally generate game levels based on the previous game level. This is not possible for many educational scenarios where exercises are handcrafted by experts. Finally, Fernandes and Levieux [15] aim to quickly adapt to players without using any prerecorded data points. To this end, they use the first 20 interactions of each new player to generate a dataset for logistic regression. While this is feasible for fast-paced games, it requires too many interactions to work with topics like math or logic, where each exercise may take minutes.

We propose a new DDA approach to address the drawbacks of the aforementioned approaches for cases where it is hard to define KCs and only a limited number of exercises and prerecorded data is available. We use an IRT model based on attributes to learn the difficulty values for all exercises based on a limited set of prerecorded data points. To quickly adapt to new students, we add a discount value to the model update to weaken IRT's assumption of constant skill.

3. APPROACH

To adjust the difficulty, we need a model that describes the student's behavior, and then a method to decide on the difficulty based on that. We describe this process in the following.

3.1 Student Model

The student model contains a set of student attributes and provides the probability of observing each possible student action. In our case, the set of possible actions is the success of or failure to solve an exercise. For our educational scenario, we find that the 4PL model [2] is a good fit. It is feasible to train on small-scale datasets. It also models the guess and slip probabilities - the chances of accidentally getting the exercise right or wrong - which is an inherent feature of the kind of exercises we work with. The original 4PL model, describing the probability of a student u_i solving an exercise q_j, is written in Equation 1.

$$p_{solve}(u_i, q_j) = c + (d - c)\frac{1}{1 + e^{-a_j(\theta_i - b_j)}} \tag{1}$$

where θ_i describes the student's ability, a_j defines the discriminatory power of the exercise, i.e. how sharply the exercise can distinguish students of different ability levels, b_j the exercise difficulty, c the probability that the user guesses the correct answer, and d the probability that the user does not slip with a wrong answer. In our scenario where all the exercises are of the same nature, we find that every exercise should have the same discriminatory power. Therefore, we learn a single value of a that applies to every exercise j. The difficulty b_j will also not be learned separately for each exercise, as each exercise in our dataset has far too few samples. Instead, we calculate b_j from the attributes of our exercises which will be explained in Section 4.1. This is done by learning weights $\vec{w}$ for the attributes:

$$b_j = a\vec{ttr}_j \cdot \vec{w} \tag{2}$$

where $a\vec{ttr}_j$ is the vector listing the attributes of the exercise, for example, the number of vertices and edges of a graph. Through this, we learn a shared set of weights for all the exercises instead of learning b_j for each exercise individually.

3.2 Training Exercise Parameters

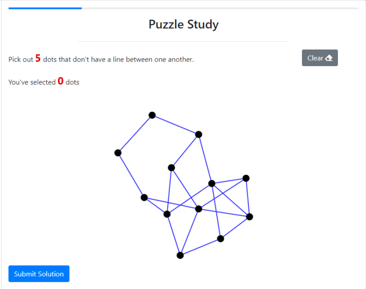
We separate the training process of our model into two steps. The first step is fitting the global parameters of the exercises on prerecorded data. The second step is to fit the student ability of each student during deployment. This two-step training is inspired by Xu et al. [38] and has been shown to work well and efficiently. The model we train in the first step is the original 4PL model. The parameters we want from this are the attribute weights $\vec{w}$ and the a, c, d values from the 4PL model introduced in Section 3.1. Naturally, we do need to include the student ability θ for the training to work, but we will not use this θ further after training. We optimize the joint maximum likelihood of all student's past actions using gradient descent, similar to Warm et al. [37]:

$$L = \sum_{(u_i, q_j, y_{solve}, t) \in \mathscr{D}} l(y_{solve}, p_{solve}(u_i, q_j)), \tag{3}$$

where the tuple (u_i, q_j, y_{solve}, t) represents the event of student u_i making an attempt on exercise q_j with success outcome $y_{solve} \in \{0, 1\}$ at time step $t \in \mathbb{N}$ and $l(\cdot, \cdot)$ is the cross-entropy loss. This describes the first step of training, which uses prerecorded data.

3.3 User Ability Update & Difficulty Adjustment

When deploying the DDA model to a new student, we fix the exercise parameters $\vec{w}, a, c$, and d learned in Section 3.2. We only learn the student's ability value θ. Every time the student finishes an exercise, we run gradient descent on all observed attempts by this student until it converges or reaches a maximum number of 1,000 iterations.


Puzzle Study

Pick out 5 dots that don't have a line between one another.

You've selected 0 dots

Figure 1: The user interface of the graph theory exercises in our study. This contains the graph, the relevant buttons, and the counter of currently selected vertices.

One caveat of the original 4PL model is that it assumes that students have an unchanging ability level. This does not reflect how students learn a new concept. To remedy this, we add a discount value $\gamma \in (0, 1)$ to our maximum likelihood function (Equation 3). With this, we weigh the past actions by γ^{T-t}, where T is the current time step and t is that action's time step. By giving less weight to past evidence, we make the change in ability level more fluid and more reliant on recent outcomes. The loss function we optimize for becomes

$$L = \sum_{(u_i, q_j, y_{solve}, t) \in \mathscr{D}, t<T} \gamma^{T-t} l(y_{solve}, p_{solve}(u_i, q_j)). \tag{4}$$

After the ability value is updated, the probability of solving p_{solve} is calculated for each exercise. The exercise with p_{solve} closest to a desired success rate is chosen and provided to the student next. For this, we need to pick the success rate that the students should get. Gonzalez-Duque et al. [11] suggest a success rate between 50% and 70%, while Klinkenberg et al. [21] suggest 75%. Therefore we tested our DDA approach using smaller pilot studies with a target success rate of 70%, 65%, and 60%. Since both 70% and 65% provided too easy exercises, we opted for 60% for our final study.

4. EXPERIMENTS

4.1 Task

For our experiments, we use an example graph theory lecture, where students are introduced to the Maximum Independent Set (MIS) of a graph (i.e., the largest set of vertices such that none of the selected vertices are adjacent to one another). It is a concept in graph theory that students need to be familiar with, therefore the setting simulates a real learning scenario. Furthermore, most people have not heard of MIS before, so it is a newly introduced concept. Finally, the definition of the MIS is simple, yet finding an MIS for a given graph is difficult. It requires an intuition that is best built through exercises. To this end, we generate a pool of 191 exercises that can be provided to the students. Figure 1 shows an example of such an exercise during our study.

4.2 Training the DDA Model

We trained our model as described in Section 3. To learn the difficulty b_j of each exercise we use the attributes $a\vec{ttr} = (|V_{MIS}|, p_{alg}, |V|, |E|, |I|)$, where $|V|$ and $|E|$ are the number of vertices and edges in the graph, $|V_{MIS}|$ is the size of the MIS, p_{alg} is the success rate of a stochastic solver algorithm on this graph (see Appendix A), and $|I|$ is the number of intersections of edges. To pre-train our DDA model, we collected data from 80 users without using DDA. For details of the training parameters and the data collection see Appendix B.

4.3 Simulation Design

Before starting the user study, we carry out experiments with simulations to verify that our algorithm works in adjusting to simulated students' behavior. We handcrafted simulated students that interact with the DDA algorithm. Our simulated students have an internal ability value. If the ability value plus a Gaussian noise is greater than the exercise's difficulty, then the attempt is a success. Otherwise, it is a failure. The ability value is increased by learning, which happens when the exercise is given at the right level of difficulty in accordance with Vygotsky's zone of proximal development [32]. The student also has a boredom and anxiety value, which increase when an exercise is too easy or too hard, respectively. Learning only occurs when the sum of boredom and anxiety values is lower than a set threshold. For details refer to the repository that contains our implementation and data [1]. To emulate our user study (see Section 4.4), the simulation starts with three fixed pre-test exercises, where the DDA algorithm updates its student model but does not choose the exercises. Then it loops through 12 training exercises that are chosen by the DDA algorithm.

4.4 User Study Design

Procedure: In our user study, the participants are presented with a sequence of MIS exercises, divided into three phases: pre-test, training, and post-test. Before the experiment, there is a questionnaire for the participant's demographics and their general interest in puzzles, computer science, and mathematics. After that, the students are provided with a tutorial that resembles the part of a graph theory lecture that introduces Maximum Independent Sets (MIS). This also includes a tutorial exercise to make sure that the participants understood the task correctly. The pre- and post-test phases are fixed and each contains one exercise of easy, medium, and hard difficulty based on a handcrafted difficulty metric (see Appendix A). The tests provide bonus payments and are used to motivate participants to practice. The training phase consists of 12 exercises where our proposed algorithm (see Section 3) runs in the background to estimate the student's ability and provides exercises that the student is estimated to have a 60% probability of solving. After the post-test, there is another free-text questionnaire asking about the task difficulty and the student's feelings about the task. For each exercise, the participant sees the graph and the number of vertices that need to be selected (Figure 1). The exercise will not be declared solved automatically once the correct vertices are chosen, but the participant has to manually click "Submit". They are told that there is a time limit on each exercise but do not know how long it is (90 seconds). This is done to reduce the sensation of time and enable flow. We display a red flag 5 seconds before the time runs out to remind them to submit the solution if they think they have one. After submitting, there is a pop-up saying if the solution was correct. The median time the experiment took was 23 minutes.

Participants & Compensation: We recruited 30 participants using the online platform Prolific. They were required to be fluent in English. For some participants, the training exercises were too difficult overall and they failed to solve more than one training exercise. Removing those participants from the analysis left us with 25 participants. The participants included 15 males and 10 females, with ages ranging from 20 to 47 years old and a mean of 29.2. Each participant was paid £3.9 for successful participation. Additionally, for each correctly solved pre- and post-test exercise, they were paid £0.1 - 0.2, depending on the time they needed. This totals up to a potential bonus of £1.2.

Research Questions & Hypotheses: The main research question for our study was whether our DDA approach adapts as intended. For this, we hypothesized that there is no significant difference between the desired success rate (60%) and the actual success rate of the participants during the training phase. As a secondary research question, we are interested in investigating if our adapted IRT model is still useful for knowledge tracing. That would be the case if there is a correlation between the ability level that the student model within our DDA approach assigns to each participant after training and their performance in the post-test. This work presents the first part of a bigger study that we preregistered online. [2] To keep the scope of this work focused on the adaptation, we will describe the results of the remaining study in future work after a thorough analysis.

5. RESULTS

5.1 Result of the Simulations

During the simulation, we used three groups of 50 simulated students where each group simulated students with a different initial ability level (low, medium, and high). To visualize how the DDA algorithm performs, we designed a handcrafted metric for the difficulty of each exercise (see Appendix A). Figure 2 shows the trajectory of the student's ability and the difficulty of the exercises provided to them in each time step. Right up from the first time step of the training phase, there was a difference in the difficulties that the different students get because of the different pre-test results. The students with higher initial abilities got harder exercises. The difficulty was slightly lower than the student's ability because we want to have a 60% success rate. As the users increased in their ability level with training, the difficulty of the exercises provided to the students also increased, showing the system can detect the change and adapt itself accordingly. Taking a look at the success rate of exercise solves during the training phase, the simulated students were able to solve 61.1% ± 12.4% of the exercises. Using a one sample t-test, we find no significant difference from 60%; t(df) = 1.08, p = 0.28, Cohen's d = 0.09. Also, DDA yielded higher learning than random and predefined difficulty curve baselines. Simulated students on DDA improved by a mean of 12.5 difficulty units between pre- and post-test, while predefined difficulty curve and random improve by 9.6 and 6.9 difficulty units, respectively.

5.2 Results of the User study

For our main research question, we wanted to verify that our approach was capable of providing exercises with a 60% success rate. Our participants successfully solved on average 7.0 ± 1.2 out of 12 training exercises, which translates to a success rate of 58 ± 10%. Using a one sample t-test, we find that the success rate was not significantly different from our aimed 60% with a very small effect size; t(df) = -0.86,

[1] `https://github.com/hcmlab/fast-dda-for-its`.

[2] `https://aspredicted.org/i6zm7.pdf`

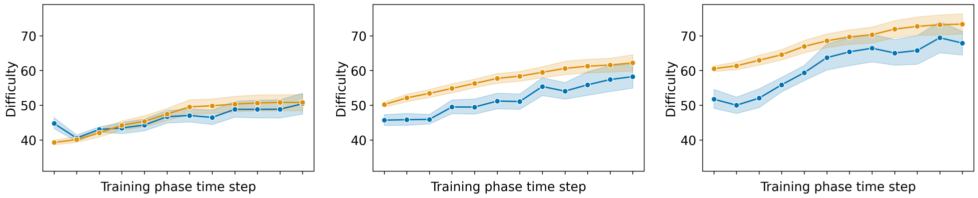


Figure 2: Mean ability level (orange) of simulated students throughout the simulation and difficulty levels (blue) provided to them by DDA. The figures show curves for different initial ability levels, low to high from left to right. The fixed pre-test exercises are not shown here, so some adaptation is visible already from the beginning. Our algorithm correctly provided exercises that are slightly below the student's ability level since we aim for a 60% success rate. The error band shows the 90% CI.

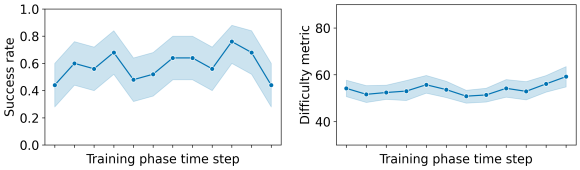


Figure 3: Mean success rate (left) and exercise difficulty (right) at each time step during the training phase of the human participants. We omit the pre- and post-test phases.

p = 0.40, Cohen's d = 0.17. For context, when collecting training data without DDA, the mean success rate was 49 ± 20%. In Figure 3 we show the mean success rate and exercise difficulty during the training phase. To see if our DDA approach can infer the student's post-test performance from the training session, we calculated the Spearman correlation between the fitted ability value θ after the training phase and the post-test score. We found significant correlations for the data collected without DDA ((r^2 = 0.599, p = 0.003) for predefined difficulty and (r^2 = 0.522, p = 0.006) for self-determined difficulty). For the data collected with DDA we found no significant correlation (r^2 = 0.144, p = 0.49).

6. DISCUSSION

The main goal of this paper was to introduce a DDA algorithm that is able to quickly adapt to students after only a few interactions. During both our simulated and human user experiments we did not find a significant difference from the desired success rate of 60%, with very small effect sizes in both experiments. This indicates that there is no large difference between our outcome and the desired success rate. For simulated users, it also did so while staying in the zone of proximal development where our simulated users learned the most. This shows that it did not simply give very easy and very hard exercises to get the given quota but actually estimated the student's chance of success for each exercise. For real users, the mean success rate hovered around 60% throughout the course of the experiment while the average difficulty increased towards the end (see Figure 3). This shows that the participants improved and that our DDA algorithm adapted to them correctly. As an example for individual students, we show and discuss the progression of two students in Appendix C. While previous user studies with many interactions managed to achieve their desired success rate [21], Schadenberg et al. [35] showed that this is not a trivial task for scenarios with limited numbers of interactions. In their user study, which used a similar amount of interactions as our study, they were not able to change the success rate compared to their baseline.

We also took a closer look at the 5 participants that we excluded from the initial evaluation because they did not get more than one exercise correct during training. Our DDA algorithm was correctly providing them with the easiest possible puzzle in each time step, but could not give them easier exercises because there just were no easier exercises in the pool. To alleviate this limitation, future applications could utilize this ability of the DDA algorithm to identify situations where students are struggling. Based on this automatic detection, a human teacher or tutor could intervene and provide further help individually.

We checked whether the participant's θ inferred during the training phase correlates with their post-test performance. We found that this correlation exists when the DDA does not choose the exercises, but not when we enforce a 60% success rate. This is in line with the findings by Eggen and Verschoor that the further the success rate is from 50%, the worse the IRT models estimate the performance [13].

Our work has some limitations. First, our approach is only suitable for problems that can be parameterized by attributes. Second, our modeling of student learning might be less nuanced than models based on KCs that can show how well the students know each KC. Finally, a direct comparison with algorithms like ELO or BKT as well as an investigation of the learning performance of the students is needed to fully grasp the contribution of our method. This was not within the scope of this work.

7. CONCLUSION & FUTURE WORK

In this work, we proposed a novel DDA algorithm for intelligent tutoring systems with a limited number of interactions and small datasets. We showed in simulations and a user study that our approach is able to achieve the desired success rate after limited interactions. In the future, we plan to evaluate how this influences the students' learning process. Furthermore, future work should investigate the potential of DDA algorithms to detect situations where students require additional support.

8. ACKNOWLEDGMENTS

This research was partially supported by the Affective Computing & HCI Innovation Research Lab between Huawei Technologies and the University of Augsburg.

References

[1] G. Abdelrahman, Q. Wang, and B. P. Nunes. Knowledge tracing: A survey. *ACM Comput. Surv.*, 55(11):224:1–224:37, 2023.

[2] M. A. Barton and F. M. Lord. An upper asymptote for the three-parameter logistic item-response model. *ETS Research Report Series*, 1981(1):i–8, 1981.

[3] K. N. Bauer, R. C. Brusso, and K. A. Orvis. Using adaptive difficulty to optimize videogame-based training performance: The moderating role of personality. *Military Psychology*, 24(2):148–165, 2012.

[4] R. Belfer, E. Kochmar, and I. V. Serban. Raising student completion rates with adaptive curriculum and contextual bandits. In *International Conference on Artificial Intelligence in Education*, pages 724–730. Springer, 2022.

[5] H. Cen, K. Koedinger, and B. Junker. Learning factors analysis–a general method for cognitive model evaluation and improvement. In *Intelligent Tutoring Systems: 8th International Conference, ITS 2006, Jhongli, Taiwan, June 26-30, 2006. Proceedings 8*, pages 164–175. Springer, 2006.

[6] H. Cen, K. Koedinger, and B. Junker. Comparing two irt models for conjunctive skills. In *Intelligent Tutoring Systems: 9th International Conference, ITS 2008, Montreal, Canada, June 23-27, 2008 Proceedings 9*, pages 796–798. Springer, 2008.

[7] B. Clement, D. Roy, P. Oudeyer, and M. Lopes. Multi-armed bandits for intelligent tutoring systems. In *Proceedings of the 8th International Conference on Educational Data Mining, EDM 2015, Madrid, Spain, June 26-29, 2015*, page 21. International Educational Data Mining Society (IEDMS), 2015.

[8] G. Corbalan, L. Kester, and J. J. Van Merriënboer. Selecting learning tasks: Effects of adaptation and shared control on learning efficiency and task involvement. *Contemporary Educational Psychology*, 33(4):733–756, 2008.

[9] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[10] M. Csikszentmihalyi. *Flow: The psychology of optimal experience*, volume 1990. Harper & Row New York, 1990.

[11] M. G. Duque, R. B. Palm, D. Ha, and S. Risi. Finding game levels with the right difficulty in a few trials through intelligent trial-and-error. In *IEEE Conference on Games, CoG*, pages 503–510, 2020.

[12] M. G. Duque, R. B. Palm, and S. Risi. Fast game content adaptation through bayesian-based player modelling. In *IEEE Conference on Games, CoG*, pages 1–8. IEEE, 2021.

[13] T. J. Eggen and A. J. Verschoor. Optimal testing with easy or difficult items in computerized adaptive testing. *Applied Psychological Measurement*, 30(5):379–393, 2006.

[14] A. E. Elo. *The Rating of Chessplayers, Past and Present*. Arco Pub., New York, 1978.

[15] W. R. Fernandes and G. Levieux. \delta -logit : Dynamic difficulty adjustment using few data points. In *Entertainment Computing and Serious Games - First IFIP TC 14 Joint International Conference, ICEC-JCSG 2019, Arequipa, Peru, November 11-15, 2019, Proceedings*, volume 11863 of *Lecture Notes in Computer Science*, pages 158–171. Springer, 2019.

[16] J. Hattie. *Visible learning for teachers: Maximizing impact on learning*. Routledge, 2012.

[17] T. Huber, S. Mertes, S. Rangelova, S. Flutura, and E. André. Dynamic difficulty adjustment in virtual reality exergames through experience-driven procedural content generation. In *IEEE Symposium Series on Computational Intelligence, SSCI 2021, Orlando, FL, USA, December 5-7, 2021*, pages 1–8, 2021.

[18] R. Hunicke. The case for dynamic difficulty adjustment in games. In *Proceedings of the International Conference on Advances in Computer Entertainment Technology, ACE*, pages 429–433, 2005.

[19] B. E. John and D. E. Kieras. The goms family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 3(4):320–351, 1996.

[20] M. M. Khajah, Y. Huang, J. P. González-Brenes, M. C. Mozer, and P. Brusilovsky. Integrating knowledge tracing and item response theory: A tale of two frameworks. In *CEUR Workshop proceedings*, volume 1181, pages 7–15. University of Pittsburgh, 2014.

[21] S. Klinkenberg, M. Straatemeier, and H. L. van der Maas. Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers & Education*, 57(2):1813–1824, 2011.

[22] V. Kodaganallur, R. R. Weitz, and D. Rosenthal. A comparison of model-tracing and constraint-based intelligent tutoring paradigms. *Int. J. Artif. Intell. Educ.*, 15(2):117–144, 2005.

[23] D. Kostons, T. van Gog, and F. Paas. Self-assessment and task selection in learner-controlled instruction: Differences between effective and ineffective learners. *Computers & Education*, 54(4):932–940, 2010.

[24] D. Leyzberg, S. Spaulding, and B. Scassellati. Personalizing robot tutors to individuals' learning differences. In *2014 9th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 423–430. IEEE, 2014.

[25] H. Moon and J. Seo. Dynamic difficulty adjustment via fast user adaptation. In *UIST '20 Adjunct: The 33rd Annual ACM Symposium on User Interface Software and Technology, Virtual Event, USA, October 20-23, 2020*, pages 13–15. ACM, 2020.

[26] O. Pastushenko. Gamification in assignments: Using dynamic difficulty adjustment and learning analytics to enhance education. In *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, CHI PLAY '19 Extended Abstracts, page 47–53, New York, NY, USA, 2019. Association for Computing Machinery.

[27] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis - A new alternative to knowledge tracing. In V. Dimitrova, R. Mizoguchi, B. du Boulay, and A. C. Graesser, editors, *Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling, Proceedings of the 14th International Conference on Artificial Intelligence in Education, AIED 2009, July 6-10, 2009, Brighton, UK*, volume 200 of *Frontiers in Artificial Intelligence and Applications*, pages 531–538. IOS Press, 2009.

[28] R. Pelánek. Applications of the elo rating system in adaptive educational systems. *Computers & Education*, 98:169–179, 2016.

[29] R. Pelánek, J. Papoušek, J. Řihák, V. Stanislav, and J. Nižnan. Elo-based learner modeling for the adaptive practice of facts. *User Modeling and User-Adapted Interaction*, 27(1):89–118, 2017.

[30] E. Poromaa. Crushing candy crush : Predicting human success rate in a mobile game using monte-carlo tree search. 2017.

[31] M. D. Reckase. Unifactor latent trait models applied to multifactor tests: Results and implications. *Journal of educational statistics*, 4(3):207–230, 1979.

[32] M. M. Rohrkemper. Self-regulated learning and academic achievement: A vygotskian view. In *Self-regulated learning and academic achievement*, pages 143–167. Springer, 1989.

[33] C. Romero, S. Ventura, E. L. Gibaja, C. Hervás, and F. Romero. Web-based adaptive training simulator system for cardiac life support. *Artificial Intelligence in Medicine*, 38(1):67–78, 2006.

[34] R. J. Salden, F. Paas, and J. J. Van Merriënboer. Personalised adaptive task selection in air traffic control: Effects on training efficiency and transfer. *Learning and Instruction*, 16(4):350–362, 2006.

[35] B. R. Schadenberg, M. A. Neerincx, F. Cnossen, and R. Looije. Personalising game difficulty to keep children motivated to play with a social robot: A bayesian approach. *Cognitive systems research*, 43:222–231, 2017.

[36] T. Schodde, K. Bergmann, and S. Kopp. Adaptive robot language tutoring based on bayesian knowledge tracing and predictive decision-making. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 128–136, 2017.

[37] T. A. Warm. Weighted likelihood estimation of ability in item response theory. *Psychometrika*, 54(3):427–450, 1989.

[38] L. Xu and M. A. Davenport. Dynamic knowledge embedding and tracing. In A. N. Rafferty, J. Whitehill, C. Romero, and V. Cavalli-Sforza, editors, *Proceedings of the 13th International Conference on Educational Data Mining, EDM 2020, Fully virtual conference, July 10-13, 2020.* International Educational Data Mining Society, 2020.

[39] S. Xue, M. Wu, J. Kolen, N. Aghdaie, and K. A. Zaman. Dynamic difficulty adjustment for maximized engagement in digital games. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 465–471, 2017.

[40] A. Yazidi, A. Abolpour Mofrad, M. Goodwin, H. L. Hammer, and E. Arntzen. Balanced difficulty task finder: an adaptive recommendation method for learning tasks based on the concept of state of flow. *Cognitive Neurodynamics*, 14(5):675–687, 2020.

[41] Y. Zhang and W.-B. Goh. Personalized task difficulty adaptation based on reinforcement learning. *User Modeling and User-Adapted Interaction*, 31(4):753–784, 2021.

Table 1: The Spearman's rank correlation coefficients between human participants' performance and the different attributes of our graph theory exercises.

Exercise Attributes	Time	Correctness
Difficulty Metric	0.2778	-0.2672
Stochastic Solve Probability	-0.2007	0.1841
Number of Vertices	0.2422	-0.2059
MIS Size	0.2541	-0.1651
Number of Edges	0.2217	-0.2427
Number of Intersections	0.1518	-0.2249

APPENDIX

A. DETAILS OF STOCHASTIC SOLVER & HANDCRAFTED DIFFICULTY METRIC

To infer the difficulty score of the generated MIS exercise, we use a stochastic solver, similar to Poromaa [30]. The stochastic solver tries to find an MIS of the provided graph by choosing vertices in a non-deterministic way. The more often this solver finds a correct solution, the easier we consider the graph to be.

The stochastic algorithm starts with no selected vertices. In each step, each free vertex (i.e., an unselected vertex without selected neighbors) is given a probability of being chosen into the set. The probability that a vertex is chosen is inversely proportional to the number of its neighbors that are also free vertices. The algorithm samples from this categorical distribution and adds one vertex to the set of selected vertices. The algorithm loops until there is no free vertex left. To verify whether a stochastic solution is maximum, we calculate the correct size of the MIS, denoted $|V_{MIS}|$, for each graph by brute force beforehand. We run the stochastic algorithm 10,000 times and count the number of successful solves to get the success rate p_{alg}.

The handcrafted difficulty metric of an exercise is $\frac{|V_{MIS}|}{p_{alg}} + |V| + |E|$, where $|V|$ and $|E|$ are the number of vertices and edges in the graph respectively. This value tries to mimic the number of elements a human needs to consider and the average required number of clicks to solve the exercise, in a similar vein to John et al. [19]. To make sure that the handcrafted difficulty metric, which we use for our evaluation (see Section 4.1), works as intended, we verified that it actually reflects the difficulty for real users. To this end, we calculated Spearman's rank correlation between each attribute of an exercise, including our difficulty metric, and the participants' solving outcome, i.e. whether the attempt was correct, and the time taken for each exercise. These correlations are shown in Table 1. Out of all the attributes we considered, our handcrafted difficulty metric correlates best with the outcome in both aspects.

B. DDA MODEL TRAINING DETAILS & HYPERPARAMETERS

To pre-train our DDA model, we used data from our pilot studies and collected additional user data without using DDA. For this purpose, we used two other methods to select the difficulty of training exercises for 30 participants each. The first is a predefined difficulty curve, where training exercises start easy and gradually get more difficult, regardless of the outcome of training. The second method is the self-determined difficulty, where the first training exercise has a medium difficulty. After each exercise, the student is asked whether he wants an exercise with the same difficulty, a more difficult one, or an easier one. The next exercise is provided accordingly. Altogether, we obtained 1,200 data points from 80 users after removing users that did not get any exercise correct during training and removing post-test exercises. Pre-training is done using 90% train and 10% validation split. We use the Adam optimizer with a learning rate 0.005, batch size 64, and 5,000 epochs. After pre-training on the prerecorded data is done, the DDA model is deployed to adapt to each student. During this adaptation, we apply a discount factor $\gamma = 0.7$ in the loss function as described in Section 3.3. Because of implementation reasons, we use gradient descent with a learning rate of 0.001 to fit the student's ability value θ.

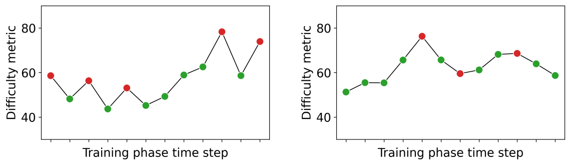


Figure 4: Progression during training of two human participants, A (left) and B (right). The y-axis shows our handcrafted difficulty metric. The dots are green for successes and red for failures.

C. EXAMPLE STUDENTS

To get an in-depth view of how our algorithm performs, we show the progression of two individual participants in Figure 4. We list the training exercises they received and plot the difficulty of each exercise according to the handcrafted difficulty metric. Student A had the easy pre-test exercise correct and thus got assigned a medium exercise at first. In the beginning, it seems like they are staying in their zone of flow which can be seen by the zigzagging between slightly too easy and too hard exercises. Then they seem to improve which our algorithm picks up on and provides harder exercises. Towards the end, the adaptation again seems to reach the student's flow zone. Student B also solved the easy pre-test correctly. After some successes, the DDA algorithm tried to give them a harder exercise but sees that the student could not work with it. After this, the difficulty stays quite level. Even when the student slips with exercises of a difficulty level they evidently solved before, the algorithm does not immediately decrease the difficulty. This seems to have been the correct procedure as student B stated in the post-questionnaire: "I felt like the difficulty level of the puzzle was just right as it made you think twice before answering.".

Discovering prerequisite relationships between knowledge components from an interpretable learner model

Olivier Allègre
Sorbonne Université, CNRS, LIP6
olivier.allegre@lip6.fr

Amel Yessad
Sorbonne Université, CNRS, LIP6
amel.yessad@lip6.fr

Vanda Luengo
Sorbonne Université, CNRS, LIP6
vanda.luengo@lip6.fr

ABSTRACT

We propose in this work a novel approach to retrieve the prerequisite structure of a domain model from learner traces. We introduce the E-PRISM framework that includes the causal effect of prerequisite relationships in the learner model for predicting the learner's performance with knowledge tracing. By studying the distribution of the learned values of each learner model parameter from synthetic data, we propose new metrics for measuring the existence, direction, and strength of a prerequisite relationship. We apply the same methodology to real-world datasets and observe promising results in retrieving the prerequisite structure of a domain model from learner traces.

Keywords

Learner modeling, prerequisite structure, data mining, Bayesian networks, knowledge tracing

1. INTRODUCTION

The prerequisite relationships, which describe dependencies between knowledge components, play a crucial role in determining the most effective instruction sequence for students. The objective of this research is to answer the following question: is it possible to propose a learner model where the parameters are enough interpretable to detect the domain model's prerequisite relationships, on top of predicting the learner performance?

We introduce the E-PRISM framework, which relies on an interpretable learner model, to analyze learners' data and detect the prerequisite structure of the domain model. We summarize our contributions in this work as follows. First, we introduce an effective and tractable method for incorporating prerequisite relationships into a continuous scale of the learning process. Second, we define new metrics for assessing the causal impact of prerequisite relationships utilizing the interpretable parameters of the E-PRISM learner model and we apply them to real-world datasets.

O. Allègre, A. Yessad, and V. Luengo. Discovering prerequisite relationships between knowledge components from an interpretable learner model. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 490–496, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115738

2. DISCOVERING THE PREREQUISITE STRUCTURE OF THE DOMAIN MODEL THROUGH LEARNER MODELING

We provide an overview of the current state-of-the-art methods for retrieving the prerequisite structure of the domain model through learner modeling. We focus specifically on the learner performance prediction models and how they are used in the literature to determine the prerequisite structure within a domain model.

2.1 Approaches in learner modeling

In the field of learner modeling a variety of algorithms can be used to predict students' performance on assessments, diagnose their strengths and weaknesses, and track their learning progress over time.

One of the popular and used methods is logistic regression, a statistical model to predict the likelihood of an event occurring given a set of predictors or independent variables. Some logistic regression algorithms, such as IRT [11] and MIRT [19], use simple features, while others, such as LFA [2], PFA [17], DAS3H [3], and Best-LR [9], use engineered and more complex features.

Besides, cognitive diagnosis algorithms model the learner's knowledge state to predict their answers. Non-temporal Bayesian network (BN) approaches, such as DINA [10], NIDA [14], and DINO [21], use BNs to compute the probability of answering correctly by modeling the learner's mastery of Knowledge Components (KCs). Bayesian Knowledge Tracing (BKT) uses BNs to track the learner's knowledge over time [5] and assumes knowledge states to be dynamic.

Deep learning techniques have been applied to learner modeling and have gained popularity due to their ability to learn and extract features from large and complex datasets automatically. Deep Knowledge Tracing (DKT) is a deep learning model for the knowledge tracing task using a neural network to learn a non-linear model of the learner's knowledge, allowing it to capture more complex patterns and make more accurate predictions [18]. Variants of DKT have been developed, but they generally only show minor performance gains compared to the original DKT model [20], except Self-Attentive Knowledge Tracing (SAKT) [16]. However, Jaeger has reported that even the more interpretable deep learning techniques are less interpretable than probabilistic graphical models such as BNs [12].

2.2 Prerequisite structure in learner models

A priori knowledge of the domain to construct a model of the prerequisite structure has been integrated into simple learner models, most of the time with Bayesian networks (BN) [4, 1]. These techniques typically involve experts using their domain knowledge to define the prerequisite relationships between the KCs through the probabilities in the networks. Also, works employ data to retrieve the conditional probabilities that rule such BNs [7].

Another approach to retrieve the prerequisite structure of the domain model is to use the predicted knowledge states of a learner over time. The idea is to use the predictions made by a learner model, which estimates the learner's knowledge state at different points in time, to infer the prerequisite relationships between the knowledge components [18, 7]. This can be done by comparing the masteries of the different knowledge components over time. The prerequisite structure of the domain model can then be determined by conducting a statistical study of these inferred states.

Finally, the work of Käser et al. is notable for its use of a Dynamic Bayesian Network (DBN) to model the effect of the prerequisite structure between knowledge components in learner models [15]. The DBN includes arcs between the variables of related KCs' mastery, which allows for modeling the causal effect of relationships between KCs. However, as the number of prerequisite KCs increases, the DBN's conditional probability distributions (CPDs) can become complex to interpret. The number of parameters grows exponentially with the number of prerequisite relationships and can be challenging to analyze. Despite this limitation, Käser's approach is a promising method for modeling the prerequisite structure in learner models, as it allows for explicitly modeling the causal effect of relationships between KCs.

3. E-PRISM: EMBEDDING PREREQUISITE RELATIONSHIPS IN STUDENT MODELING

In this research work, we introduce a new student modeling framework called E-PRISM (for **E**mbedding **P**rerequisite **R**elationships in **S**tudent **M**odeling). The E-PRISM domain model supposes a decomposition of the domain knowledge into Knowledge Components (KCs). The E-PRISM learner model assumes the learner knowledge defined as the binary masteries of each KC in the domain model. Predictions about learners' knowledge state and performance are made from data on the learner's interactions with learning systems.

3.1 Overview of the E-PRISM learner model

The learner model in E-PRISM is a knowledge-tracing model that considers variables for the mastery of several KCs of the domain model. Knowledge tracing is performed through a dynamic Bayesian network (DBN) which models the mastery of KCs over time. The DBN leverages the causal effect of the learning process and the causal effect of the prerequisite relationships to infer learners' knowledge states at any time.

E-PRISM has a key feature that sets it apart from other student modeling frameworks. It utilizes ICI-based conditional probability distributions (CPDs) [8] to model the causal effects of the learning process and the prerequisite relationships on the KC mastery at each timeslice. This defines KC mastery variables as deterministic functions of variables representing the independent causal effects that influence them. We represent the part of the DBN associated with the mastery of a KC $\mathfrak{X}$ at a time $t > 0$ in Figure 1.

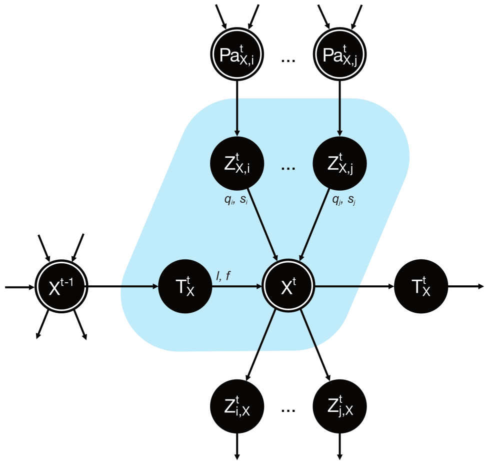


Figure 1: Noisy-AND gate of $\mathfrak{X}$ and its Markov blanket in the DBN of E-PRISM. The Noisy-AND gate is colored blue. It is composed of a variable X^t for KC mastery, defined as an AND function of auxiliary variables representing the causal effect of both its learning process and the mastery of its prerequisite KCs. The auxiliary variables are T, representing the causal effect of learning and forgetting on $\mathfrak{X}$ mastery, and Z_i for each $\mathfrak{X}$ prerequisite, representing the causal effect of the i-th prerequisite mastery on $\mathfrak{X}$ mastery. $Pa^t_{\mathfrak{X},i}$ is the variable associated with the mastery of the i-th $\mathfrak{X}$ prerequisite.

The DBN is composed of Noisy-AND gates for each KC and each timeslice. We represent a toy example of the DBN in Figure 2. The parameters of the DBN are learned with the Monte-Carlo Expectation-Maximization (MCEM) algorithm [23]. The MCEM algorithm is a variant of the Expectation-Maximization (EM) algorithm [6]. It considers the expectations of the E-step to be approximated with a Monte-Carlo sampling, which is the Blocking Gibbs sampling (BGS) [13] in our research. MCEM with BGS allows for a converging and tractable parameter learning of the learner model in E-PRISM.

3.2 Interpretability of parameters

ICI-based CPDs rely on a pair of parameters for each causal effect. In the E-PRISM learner model, there are parameters associated with the learning process, namely $(l_{\mathfrak{X}}, f_{\mathfrak{X}})$ for each KC $\mathfrak{X}$, and parameters associated with the prerequisite relationship, namely for $(q_{\mathfrak{X},i}, s_{\mathfrak{X},i})$ each prerequisite i of each KC $\mathfrak{X}$. $l_{\mathfrak{X}}$ and $f_{\mathfrak{X}}$ parameters are the probabilities of learning and forgetting $\mathfrak{X}$. $q_{\mathfrak{X},i}$ is the probability that the i-th prerequisite of $\mathfrak{X}$ is not sufficient to master $\mathfrak{X}$. On the other hand, $s_{\mathfrak{X},i}$ is the probability the i-th prerequisite of $\mathfrak{X}$ is not necessary to master $\mathfrak{X}$. These interpretable parameters allow for a clear understanding of the causal effects of the learning process and prerequisite relationships on the

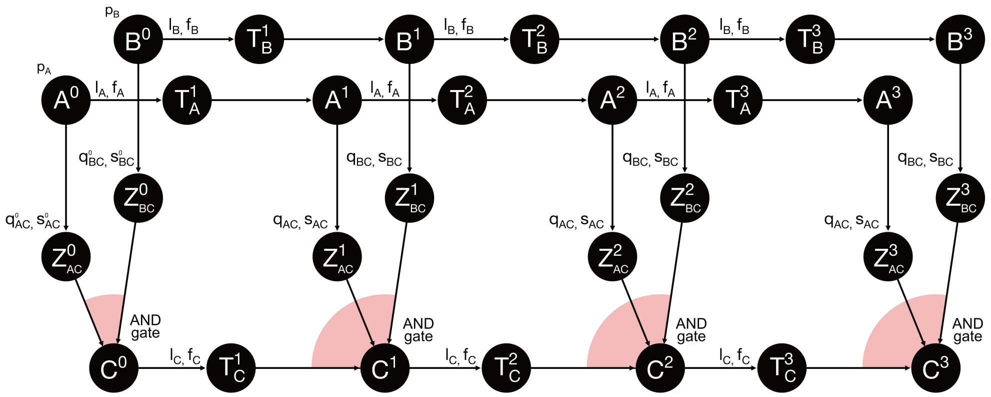


Figure 2: Example of the DBN that encodes the learner's knowledge state and considers a domain model $\{\mathfrak{A}, \mathfrak{B}, \mathfrak{C}\}$ with prerequisite relationships $\mathfrak{A} \rightarrow \mathfrak{C}$ and $\mathfrak{B} \rightarrow \mathfrak{C}$.

learner's performance. E-PRISM allows for the identification and understanding of the prerequisite structure of the domain model, which is a key focus of our research.

3.3 Metrics from E-PRISM

First, we highlight the gain of performance induced by the presence of an effective prerequisite relationship in the E-PRISM learner model. We wonder if the difference between the Root Mean Squared Error (RMSE) values obtained from different E-PRISM learner models depends on their prerequisite structure. We generate three synthetic datasets $\mathcal{D}_\varnothing$, $\mathcal{D}_{\text{weak}}$, and $\mathcal{D}_{\text{strong}}$. $\mathcal{D}_\varnothing$ is generated from an E-PRISM learner model considering no prerequisite relationship between $\mathfrak{A}$ and $\mathfrak{B}$. $\mathcal{D}_{\text{strong}}$ is generated from an E-PRISM learner model that considering a strong prerequisite relationship $\mathfrak{A} \rightarrow \mathfrak{B}$. $\mathcal{D}_{\text{weak}}$ is generated from an E-PRISM learner model considering a weak prerequisite relationship $\mathfrak{A} \rightarrow \mathfrak{B}$. By generating these synthetic datasets, we will be able to study the performance of the E-PRISM framework in different scenarios where the prerequisite relationship between $\mathfrak{A}$ and $\mathfrak{B}$ is varied. We learn the parameters of three E-PRISM learner models, namely $e\Delta_\varnothing$, $e\Delta_{\mathfrak{A}\rightarrow\mathfrak{B}}$, and $e\Delta_{\mathfrak{B}\rightarrow\mathfrak{A}}$. $e\Delta_\varnothing$ assumes no prerequisite relationship, while $e\Delta_{\mathfrak{A}\rightarrow\mathfrak{B}}$ and $e\Delta_{\mathfrak{B}\rightarrow\mathfrak{A}}$ respectively assume $\mathfrak{A} \rightarrow \mathfrak{B}$ and $\mathfrak{B} \rightarrow \mathfrak{A}$. We run 1000 simultaneous instances of the MCEM algorithm, with parameters $N_{\text{Gibbs}} = 10$ and $M = 0$, to perform E-PRISM parameter learning. The full synthetic dataset is used as a training dataset. We report the RMSE values obtained from parameter learning in Table 1.

Table 1: Best RMSE values computed by comparing E-PRISM predictions with the entire data that considers a strong prerequisite relationship. Parameter learning of E-PRISM models is also realized with the full dataset.

Method	RMSE on $\mathcal{D}_{\mathfrak{A}\rightarrow\mathfrak{B},\text{strong}}$
$e\Delta_\varnothing$	0.353
$e\Delta_{\mathfrak{A}\rightarrow\mathfrak{B}}$	**0.327**
$e\Delta_{\mathfrak{B}\rightarrow\mathfrak{A}}$	0.394

We assume the presence of an effective prerequisite relationship in the E-PRISM learner model enhances the model's performance. Thus, to study a prerequisite relationship $\mathfrak{A} \rightarrow \mathfrak{B}$, we can compare the performance of $e\Delta_{\mathfrak{A}\rightarrow\mathfrak{B}}$, the E-PRISM learner model that considers the relationship $\mathfrak{A} \rightarrow \mathfrak{B}$, and $e\Delta_\varnothing$, the model with no prerequisite relationship. We define the *LePPED* (for Learner Performance Prediction Error Difference) metric to identify the existence and the direction of the prerequisite relationship. We compute the relative difference between their RMSE value obtained after learning parameters. LePPED is computed in Equation (1). It senses the direction of the prerequisite relationship between two KCs.

$$\mathit{LePPED}(\mathfrak{A} \rightarrow \mathfrak{B}) = \frac{1}{K} \frac{(\text{RMSE of } e\Delta_\varnothing - \text{RMSE of } e\Delta_{\mathfrak{A}\rightarrow\mathfrak{B}})}{\text{RMSE of } e\Delta_\varnothing} \tag{1}$$

where K is a normalizing constant.

LePPED($\mathfrak{A} \rightarrow \mathfrak{B}$) is a measure for the existence of the prerequisite relationship, as it indicates how better the E-PRISM model performs by considering $\mathfrak{A} \rightarrow \mathfrak{B}$. *LePPED* ranges from -1 (very unlikely there exists a relationship $\mathfrak{A} \rightarrow \mathfrak{B}$) to 1 (very likely there exists a relationship $\mathfrak{A} \rightarrow \mathfrak{B}$).

Upon analyzing the distributions of the E-PRISM parameter learned values, we observed shifts in the value of the parameter when the direction of an effective prerequisite relationship is reversed. We introduce a custom metric *CPVD* (for Comparing Peak Values of the Distribution) computed by comparing the peak values of the learned parameter distributions. *CPVD* is defined in Equation (2).

$$\begin{aligned} CPVD(\mathfrak{A} \rightarrow \mathfrak{B}) = \frac{1}{6} \Big(& \mathbb{1}(l_{\mathfrak{A}}^{\mathfrak{A}\rightarrow\mathfrak{B}} > l_{\mathfrak{A}}^{\mathfrak{B}\rightarrow\mathfrak{A}}) + \mathbb{1}(l_{\mathfrak{B}}^{\mathfrak{A}\rightarrow\mathfrak{B}} < l_{\mathfrak{B}}^{\mathfrak{B}\rightarrow\mathfrak{A}}) \\ & + \mathbb{1}(f_{\mathfrak{A}}^{\mathfrak{A}\rightarrow\mathfrak{B}} < f_{\mathfrak{A}}^{\mathfrak{B}\rightarrow\mathfrak{A}}) + \mathbb{1}(f_{\mathfrak{B}}^{\mathfrak{A}\rightarrow\mathfrak{B}} > f_{\mathfrak{B}}^{\mathfrak{B}\rightarrow\mathfrak{A}}) \\ & + \mathbb{1}(q^{\mathfrak{A}\rightarrow\mathfrak{B}} > q^{\mathfrak{B}\rightarrow\mathfrak{A}}) + \mathbb{1}(s^{\mathfrak{A}\rightarrow\mathfrak{B}} < s^{\mathfrak{B}\rightarrow\mathfrak{A}}) \Big) \end{aligned} \tag{2}$$

where $\mathbb{1}$ is the identity function.

CPVD is an indicator of the existence and the direction of the prerequisite relationship. It ranges from 0 to 1. The

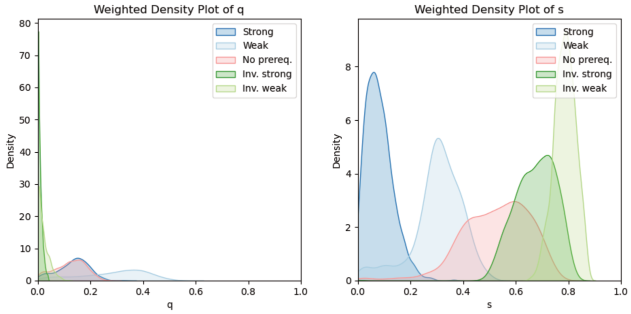


Figure 3: Distribution of the values of prerequisite parameters obtained from training on synthetic data.

greater $CPVD(\mathfrak{A} \rightarrow \mathfrak{B})$, the most likely the existence of the $\mathfrak{A} \rightarrow \mathfrak{B}$ relationship.

Finally, we benefit from the enhanced interpretability allowed by ICI-model CPDs in the E-PRISM learner model. We observe the distribution of the learned values of q and s parameters in the different situations for the E-PRISM parameter learning procedure. Specifically, we study E-PRISM learner models that either assume the correct or the wrong direction of the prerequisite relationship $\mathfrak{A} \rightarrow \mathfrak{B}$, which is expressed in the data strongly (through $\mathcal{D}_{\text{strong}}$) or weakly (through $\mathcal{D}_{\text{strong}}$).

Based on these previous observations, we propose a novel metric based on the distribution of the s parameter learned values. This second metric $\mathcal{N}ec$ is calculated by determining the proportion of learned values of s lower than 0.2 obtained in all the runs of parameter learning. It stands for the strength of the prerequisite relationship, according to the interpretation of the s parameter. The closer to 1 the value of $\mathcal{N}ec$, the stronger the prerequisite relationship between the two considered KCs.

$$\mathcal{N}ec = \frac{1}{K}\frac{\text{Number of learned parameter values lower than 0.2}}{\text{Total number of learned parameter values}} \tag{3}$$

with K a normalizing constant.

By combining these three metrics, we should be able to gain a deeper understanding of the interpretability of the E-PRISM learned parameters, and how they can be employed to retrieve the prerequisite structure (existence, direction, and strength) of the domain model in E-PRISM.

4. DISCOVERY OF THE PREREQUISITE STRUCTURE FROM REAL-WORLD DATA

4.1 Method

We study real-world data to evaluate the generalizability of the proposed metrics for measuring the existence, direction, and strength of prerequisite relationships.

We evaluate the capacity of our model to search for the existence, direction, and strength of prerequisite relationships in the *ASSISTments12*, *Eedi2020*, and *Kartable* datasets. *ASSISTments12* is issued from the ASSISTment system, with a relatively coarse granularity of KCs. *Eedi2020* was released as part of a NeurIPS2020 challenge and is issued from the Eedi system. *Kartable* is provided by Kartable and is not freely available.

We focus on the study of pairs of KCs because of tractability issues of E-PRISM with larger domain models. We consider the sub-datasets restricted to pairs of KCs and restrict each sub-dataset to learner traces from students that trained both KCs. Specifically, we have selected the 6 pairs of KCs with the highest number of learners transactions. Selected pairs of KCs are listed in Table 4 in Appendix A. Additionally, we only consider seven transactions per learner in the parameter learning procedure to ensure its tractability.

4.2 Study of the proposed metrics

We wonder how the metrics relate to the prerequisite structure of the domain model with real-world data. We report metrics' values for each selected pair of KCs in Table 2.

Some of the relationships with high custom metric scores are prerequisite relationships according to common knowledge. In particular, relationships between addition KC and multiplication KC are greatly represented. The ordering of metric values can be interpreted as a prerequisite relationship strength. Metrics *CPVD* and $\mathcal{N}ec$ show great performance for relationships *Determine if a real number is a root of a quadratic polynomial* → *Give the roots of a quadratic polynomial*, *Give the roots of a quadratic polynomial* → *Give the sign chart of a quadratic polynomial*, and *Addition and Subtraction Positive Decimals* → *Multiplication and Division Positive Decimals*. We can clearly observe that these detected prerequisite relationships, thanks to the *CPVD* and $\mathcal{N}ec$ metrics, are coherent with the mathematics domain knowledge. Nevertheless, we remark that there is also a relationship suggesting that *Multiplication and Division Integers* is a requirement of *Addition and Subtraction Integers*. These relationships should be submitted for the approval of experts in the domain.

4.3 Relative agreement between metrics

We study the relative agreement between introduced metrics for asserting the correctness of the inferred prerequisite structure. To do so, we compute the Cohen kappa [22] between the metric predictors. For each sub-dataset, we evaluate the reliability between metrics on the existence and direction of the corresponding prerequisite relationship.

For every KCs $\mathfrak{A}$ and $\mathfrak{B}$, we define predictors on the existence of the prerequisite relationship $\mathfrak{A} \rightarrow \mathfrak{B}$ from each metric by checking if they are positive. Similarly, predictors for the correct direction of the prerequisite relationship are introduced by comparing the metric value of both directions of the relationship between $\mathfrak{A}$ and $\mathfrak{B}$. We also introduce a predictor that combines the two conditions, and we present the results in Table 3.

We observe that the predictors of the existence of the prerequisite relationship give different results depending on the employed metric. The predictors for the direction of the prerequisite relationships grossly agree with each other, es-

Table 2: Scores of the metrics *LePPED*, *CPVD*, and $\mathcal{N}ec$ on relationships that have been predicted as prerequisites according to the *CPVD* and $\mathcal{N}ec$ predictors.

Order	Relationship	*LePPED*	Relationship	*CPVD*	Relationship	$\mathcal{N}ec$
1	ASI → ASF	1	ASPD → MDPD	1	Root → Solve	1
2	Chart → Solve	0.85	Solve → Chart	1	MMD → MAS	0.89
3	ASI → MDI	0.72	Root → OR	0.92	Solve → CF	0.89
4	Root → OR	0.56	Root → Solve	0.92	Solve → Chart	0.89
5	Solve → Chart	0.54	MDI → ASI	0.83	ASF → DF	0.78
6	ASF → DF	0.49	ASI → ASF	0.83	E → ASI	0.78
7	MMD → MAS	0.45	FHCF → MLCM	0.83	MAS → MMD	0.78
8	PNPF → FHCF	0.44	PNPF → MLCM	0.83	ASPD → MDPD	0.67
9	VNP → MMD	0.42	VNP → MMD	0.75	MPDP → ASPD	0.67
10	MDI → ASI	0.34	ASF → DF	0.58	FHCF → MLCM	0.67
11	OR → Root	0.33	FHCF → PNPF	0.58	PNPF → MLCM	0.67
12	MMD → VNP	0.29	MAS → VNP	0.58	Root → OR	0.56
13	DF → ASF	0.28	Chart → CF	0.58	CF → Chart	0.56
14	ASF → MF	0.23	ASF → MF	0.5	ASF → MF	0.44
15	MAS → MMD	0.23	ASI → E	0.42	ASI → ASF	0.44

Table 3: Cohen kappa values obtained from measuring the agreement of metrics *LePPED*, *CPVD*, and $\mathcal{N}ec$ on the existence and direction of the prerequisite relationships.

		Existence	Direction	Ex. + Dir.
LePPED	*CPVD*	0.133	0.325	0.111
LePPED	$\mathcal{N}ec$	−0.071	0.55	0.117
CPVD	$\mathcal{N}ec$	0.053	0.55	0.778

pecially *LePPED* with $\mathcal{N}ec$ and *CPVD* with $\mathcal{N}ec$. Finally, when considering the two conditions in the predictor, we observe a strong agreement between *CPVD* and $\mathcal{N}ec$, *CPVD* and $\mathcal{N}ec$ then suggest the same relationships to be part of the prerequisite structure of the domain. On the other hand, we observe a weak agreement (near random) between *LePPED* and the other metrics.

This result suggests that RMSE is not sufficient to infer the prerequisite relationships from data, even if it can be interpreted as a first filter to determine the existence of the prerequisite structure with *LePPED*. Nevertheless, even if the relevance of *CPVD* and $\mathcal{N}ec$ have been confirmed by the results, they should be compared with the predictions of experts, to assess that the joint agreement between *CPVD* and $\mathcal{N}ec$ indeed corresponds to the correct prerequisite structure.

5. CONCLUSIONS AND PERSPECTIVES

In conclusion, this work presents a novel approach for leveraging the causal effect of prerequisite relationships to infer students' knowledge state over time. The E-PRISM framework, which utilizes Dynamic Bayesian Networks (DBNs) to predict student performance, is based on a set of interpretable parameters that sense the causal effect of the learning process and the structure of prerequisite relationships in a specific domain. Our study demonstrates the ability of these parameters to compute metrics, such as *CPVD* and $\mathcal{N}ec$, which can infer the existence, direction, and strength of prerequisite relationships. Our results, applied to the domain of mathematics, indicate the existence of common knowledge prerequisite relationships. However, further research is necessary to verify the effectiveness of these predictions by examining each inferred relationship from an expert's point of view. In summary, this work presents a promising approach for inferring prerequisite relationships in educational data mining from analyzing an interpretable learner model.

6. REFERENCES

[1] C. Carmona, E. Millán, J.-L. Pérez-de-la Cruz, M. Trella, and R. Conejo. Introducing prerequisite relations in a multi-layered bayesian student model. In *International conference on user modeling*, pages 347–356. Springer, 2005.

[2] H. Cen, K. Koedinger, and B. Junker. Learning factors analysis–a general method for cognitive model evaluation and improvement. In *International conference on intelligent tutoring systems*, pages 164–175. Springer, 2006.

[3] B. Choffin, F. Popineau, Y. Bourda, and J.-J. Vie. Das3h: modeling student learning and forgetting for optimally scheduling distributed practice of skills. *arXiv preprint arXiv:1905.06873*, 2019.

[4] C. Conati, A. Gertner, and K. Vanlehn. Using bayesian networks to manage uncertainty in student modeling. *User modeling and user-adapted interaction*, 12(4):371–417, 2002.

[5] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[7] M. C. Desmarais, P. Meshkinfam, and M. Gagnon. Learned student models with item to item knowledge structures. *User Modeling and User-Adapted Interaction*, 16(5):403–434, 2006.

[8] F. J. Dıez and M. J. Druzdzel. Canonical probabilistic models for knowledge engineering. *UNED, Madrid, Spain, Technical Report CISIAD-06-01*, 2006.

[9] T. Gervet, K. Koedinger, J. Schneider, T. Mitchell,

et al. When is deep learning the best approach to knowledge tracing? *Journal of Educational Data Mining*, 12(3):31–54, 2020.

[10] E. H. Haertel. Using restricted latent class models to map the skill structure of achievement items. *Journal of Educational Measurement*, 26(4):301–321, 1989.

[11] R. K. Hambleton, H. Swaminathan, and H. J. Rogers. *Fundamentals of item response theory*, volume 2. Sage, 1991.

[12] M. Jaeger. Learning and Reasoning with Graph Data: Neural and Statistical-Relational Approaches. In *International Research School in Artificial Intelligence in Bergen (AIB 2022)*, volume 99 of *Open Access Series in Informatics (OASIcs)*, pages 5:1–5:42, 2022.

[13] C. S. Jensen, U. Kjærulff, and A. Kong. Blocking gibbs sampling in very large probabilistic expert systems. *International Journal of Human-Computer Studies*, 42(6):647–666, 1995.

[14] B. W. Junker and K. Sijtsma. Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, 25(3):258–272, 2001.

[15] T. Käser, S. Klingler, A. G. Schwing, and M. Gross. Dynamic bayesian networks for student modeling. *IEEE Transactions on Learning Technologies*, 10(4):450–462, 2017.

[16] S. Pandey and G. Karypis. A self-attentive model for knowledge tracing. In *12th International Conference on Educational Data Mining, EDM 2019*, pages 384–389. International Educational Data Mining Society, 2019.

[17] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis–a new alternative to knowledge tracing. In *Artificial Intelligence in Education*, pages 531–538. IOS Press, 2009.

[18] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. *Advances in neural information processing systems*, 28, 2015.

[19] M. D. Reckase. Multidimensional item response theory models. In *Multidimensional item response theory*, pages 79–112. Springer, 2009.

[20] R. Schmucker, J. Wang, S. Hu, and T. M. Mitchell. Assessing the performance of online students–new data, new approaches, improved accuracy. *Journal of Educational Data Mining*, 14(1):1–45, 2022.

[21] J. L. Templin and R. A. Henson. Measurement of psychological disorders using cognitive diagnosis models. *Psychological methods*, 11(3):287, 2006.

[22] S. M. Vieira, U. Kaymak, and J. M. Sousa. Cohen's kappa coefficient as a performance measure for feature selection. In *International conference on fuzzy systems*, pages 1–8. IEEE, 2010.

[23] G. C. Wei and M. A. Tanner. A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, 85(411):699–704, 1990.

APPENDIX

A. KNOWLEDGE COMPONENTS IN THE REAL-WORLD SUB-DATASETS

Table 4: Studied couples of knowledge components for each real-world dataset

Dataset	$\mathfrak{A}$	$\mathfrak{B}$
ASSISTments12	Addition and Subtraction Integers (ASI)	Multiplication and Division Integers (MDI)
ASSISTments12	Addition and Subtraction Fractions (ASF)	Multiplication Fractions (MF)
ASSISTments12	Addition and Subtraction Integers (ASI)	Addition and Subtraction Fractions (ASF)
ASSISTments12	Addition and Subtraction Positive Decimals (ASPD)	Multiplication and Division Positive Decimals (MDPD)
ASSISTments12	Addition and Subtraction Fractions (ASF)	Division Fractions (DF)
ASSISTments12	Addition and Subtraction Integers (ASI)	Exponents (E)
Eedi2020	Factors and Highest Common Factor (FHCF)	Multiples and Lowest Common Multiple (MLCM)
Eedi2020	Factors and Highest Common Factor (FHCF)	Prime Numbers and Prime Factors (PNPF)
Eedi2020	Multiples and Lowest Common Multiple (MLCM)	Prime Numbers and Prime Factors (PNPF)
Eedi2020	Volume of Non-Prisms (VNP)	Mental Multiplication and Division (MMD)
Eedi2020	Volume of Non-Prisms (VNP)	Mental Addition and Subtraction (MAS)
Eedi2020	Mental Addition and Subtraction (MAS)	Mental Multiplication and Division (MMD)
Kartable	Determine the canonical form of a quadratic polynomial (CF)	Give the roots of a quadratic polynomial (Solve)
Kartable	Determine if a real number is a root of a quadratic polynomial (Root)	Find an obvious root for a quadratic polynomial (OR)
Kartable	Give the roots of a quadratic polynomial (Solve)	Determine if a real number is a root of a quadratic polynomial (Root)
Kartable	Determine the canonical form of a quadratic polynomial (CF)	Give the sign chart of a quadratic polynomial (Chart)
Kartable	Give the roots of a quadratic polynomial (Solve)	Give the sign chart of a quadratic polynomial (Chart)
Kartable	Find an obvious root for a quadratic polynomial (OR)	Calculate the discriminant of a quadratic polynomial given in the expanded form (D)

"Can we reach agreement?": A context- and semantic-based clustering approach with semi-supervised text-feature extraction for finding disagreement in peer-assessment formative feedback. *

M Parvez Rashid, Divyang Doshi, Sai Venkata Vinay, Qinjin Jia, Edward F. Gehringer
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
{mrashid4, ddoshi2, ssamudr, qjia3, efg}@ncsu.edu

ABSTRACT

In the process of review for assessing a piece of work, agreement or consensus among reviewers is vital to review quality. As classroom peer assessments are undertaken by naïve peers, disagreement among peer assessors can confuse the assessees and lead them to question the review process. Although there are methods like inter-rater reliability (IRR) to measure disagreement in summative feedback, in the authors' knowledge, there is no method for finding disagreements within formative feedback. It may take more time and effort for the instructor to review the feedback to find disagreements than it would to simply perform an expert review without involving peer assessors. An automated method can help locate disagreements among reviewers. In this work, we used a clustering algorithm and NLP techniques to find disagreement in formative feedback. As the review comments are related by context and semantics, we implemented a semi-supervised approach to fine-tune the SentenceTransformer model to capture the context and semantics-based relation among the review texts, which in turn improved the comment clustering performance.

Keywords

Peer-review, disagreement, NLP, SentenceTransformer, fine-tuning, clustering

1. INTRODUCTION

Peer review has long been an effective component of students' learning experience [15]. Previous studies showed that assessment by student peers could be as accurate as assessment by the instructor [14]. Not only do students learn from reviews they receive, but they learn even more from providing feedback [11, 2, 9, 5, 13]. To make the peer-assessment process more accurate and unbiased, each artifact is generally anonymized and reviewed separately by multiple reviewers [4]. Since peer reviewers assess fewer artifacts than the instructor, they can afford to spend more time on each [1]. However, peer reviewers do not always agree with each other's reviews. In Table 1, we have shown review comments on a piece of work where reviewers had incoherent opinions.

Though it is important for reviews to be coherent, to our knowledge, no classroom peer review process implements a meta-review round to find disagreements among the reviewers. One reason is that in the peer-review process, the number of reviews can be overwhelming for an instructor to meta-review, causing far more trouble than simply reviewing the artifacts themselves. For example, r reviewers review s students for c items, makes $r \times s \times c$ reviews to meta-review. An efficient way to identify disagreements is by implementing cutting-edge NLP methods to group the reviews expressing similar opinions together and locating the disagreements using a clustering algorithm. However, grouping the peers' comments using a clustering algorithm is not a straightforward task. Peer reviewers are often given a rubric [12] and in ideal cases, reviewers are expected to find the same issues in a piece of work, which makes the review texts semantically similar. Empirically we observed comments expressing disagreement might contain similar words and structure, or conversely, similar ideas may be expressed with completely different words. For example, in response to a rubric item, "If there are functions in the agent controller, are they handling one and only one functionality?" two peer reviewers' comments on the same piece of work are, "All functions are handling one to one functionality" and "They can handle multiple functionalities." These two comments are semantically very similar but clearly, the reviewers are in disagreement. It makes a difficult case for a state-of-the-art language model to distinguish the difference. The accuracy of a text clustering model depends on the feature vectors of the texts, i.e., similar texts should be represented as similar feature vectors [10]. SBERT is a current state-of-the-art sentence feature embedding model that is designed to be fine-tuneable for a downstream dataset.

M. P. Rashid, D. Doshi, S. V. Vinay, Q. Jia, and E. F. Gehringer. "can we reach agreement?": A context- and semantic-based clustering approach with semi-supervised text-feature extraction for finding disagreement in peer-assessment formative feedback. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th Inter-national Conference on Educational Data Mining*, pages 497–501, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115724

Table 1: Table shows four peer-reviewers' comments on a piece of work following a rubric item. Three of the reviewers are in agreement, and one reviewer disagrees.

Rubric Item	Student	Reviewers	Review Comments
Is the UI of the application neat and logical?	Student_1	Rev_1	UI seems awesome, but lacks functionality and features.
		Rev_2	Yes, Nav bar is very clearly implemented.
		Rev_3	The UI is not particularly neat and logical.
		Rev_4	UI is neat and I particularly liked the navigation bar.

In this study, we first compare the performance of four sentence-feature-embedding methods and pick the best method to fine-tune the model. We finally compare the clustering result using feature vectors of the pre-tarined and fine-tuned sentence embedding models. While implementing these approaches, we will address three research questions:

- **RQ1: Which pre-trained feature extraction methods for context and semantically related sentences work best?**
- **RQ2: Can we fine-tune and improve the pre-trained SBERT model's sentence-feature extraction for our context-dependent review text using a semi-supervised approach?**
- **RQ3: Does improving the sentence feature extraction method improve clustering performance?**

In this study, by "Disagreement" in peer assessors' feedback comments, we mean i) Comments that are opposing each other and ii) Comments that relate to disparate issues. Comments that agree partially with another comment are not considered in to be in disagreement.

2. RELATED WORK

Hiray et al. [7] showed that neural network models can be used to identify disagreement in online discussions. Instead of hand-crafted feature extraction they implemented a Siamese inspired neural network architecture to generate feature embedding of the texts. Guan et al. [6] discussed different text clustering approaches and found that clustering algorithms' performance depends on the quality of the feature vectors. Peer-review data in the educational environment is less available than product reviews or social-media text. The length of peer assessments is often similar in length to product reviews. Studying methods used to analyze short texts will give us some idea about analyzing peer-review texts. Jinarat et al. [8] identified that a major characteristics of short texts (e.g. facebook comments and post, tweeter text, news headline, product review etc.) is lack of context information and the presence of much jargon and abbreviations. These affect the performance of traditional text-clustering algorithms.

3. METHOD

This section describes the data collection process, dataset construction, and methods we used for the study.

3.1 Collecting Formative Feedback

We acquired data for this study from the Object Oriented Design and Development course at North Carolina State University for a period of three semesters (Spring 2021, Fall 2021, and Spring 2022). Before the review process started, students were shown examples of how to write quality feedback. The assignments were submitted and reviewed using the Expertiza system. We collected formative feedback comments from the assignment named "Program 2". The peer reviewers wrote the review comments in response to 201 different rubric items. All the reviewers' and reviewees' identities were anonymized before the feedback comments were collected for analysis, so the author of the assignment or the review comment could not be determined.

3.2 Creating the Datasets:

We prepared three datasets from the review comments we collected over the three semesters. The three datasets are as follows:

1. **Sentence-Embedding-Test Dataset:** This dataset consists of 3,000 annotated pairs of review comments. The comment pair is annotated with "1" if the two review comments express a similar idea (agreement) and "0 " if they express a different idea (disagreement). These annotations were done by five experts who are familiar with the Program 2 assignment, including its rubric and review comments. Table 2 shows an example of the dataset.
2. **Fine-Tuning Dataset:** This dataset consists of 11,000 pairs of review comments. They are not initially annotated. We used this dataset for the semi-supervised approach to train the model. We annotated 1,600 pairs during the fine-tuning phase of the sentence-embedding generator model.
3. **Clustering-Test Dataset:** This dataset consists of 1,000 review comments for measuring clustering-algorithm performance quality. In this dataset we grouped all the review comments following the same rubric item on the same piece of work.

3.3 Sentence Embedding

Sentence embeddings are a way of representing different-length sentences with fixed-size vectors of numbers. In this study, we compared the performance of four sentence-embedding methods using accuracy on the Sentence-Embedding-Test Dataset.

Global Vectors (GloVE) is a word vectorization technique used to convert natural language text to feature vectors that are suitable for machine-learning models to process. GloVe incorporates the local statistics of a word in a sentence as well as the global occurrence of the word in the document.

Pre-trained Bidirectional Encoder Representations from Transformers (BERT) model produces word embeddings that has

Table 2: Table shows a sample of Sentence-Embedding-Test dataset with paired comments, labeled for agreement (Label ”1”) and disagreement (Label ”0”) in two comments. Each pair of comments is on the same piece of work following the same rubric item

	Comment1	Comment2	Label
1	Application to properties should be when reviewing a property, and the application deployment is crashing hence not able to actibely test	All the required fields of student are enforced to be non-null.	0
2	Any required attributes can be null in property class.	There is validation check for all necessary attributes	0
3	New property creation throws some application error, cannot test.	Could not apply to a property, showing a crashing application.	1
4	Yes, validations seem to be enforced	All fields were appropriately validated.	1

shown great success in finding contextual and semantic relations among words in a sentence. It is a multi-layer bidirectional model based on the encoder mechanism of the transformer model. BERT learns the contextual relation of each word by considering the other words in both directions in a sentence. We can get embeddings from BERT by using a mean-pooling method that averages the feature vectors of each word or by the [CLS] [3] token available at the first position of the BERT sentence embedding output.

Sentence-BERT (SBERT) utilizes a Siamese Neural Network, where the neural network consists of two identical subnetworks. The identical subnetworks have the same parameters and weights. The parameter updating is also mirrored in both sub-networks. This model produces sentence embedding in a way that the semantically similar sentences have a very high cosine similarity. Unlike BERT, the Siamese network does not require every possible pair combination to find semantic similarity in sentences. As a result, the computation time is reduced from $O(n^2)$ to $O(n)$.

3.4 Active Learning

For this study, we used a semi-supervised approach known as active learning to fine-tune the SBERT model. The key idea for an active-learning algorithm is that a machine-learning model can run faster and with less labeled data if it can choose the data from which the model needs to learn. During the iterative process of training models, an expert need to annotate only the samples the model is uncertain of, and the model can be trained with these annotations. Continuing this approach iteratively helps the model learn faster with few annotated samples.

3.5 Choosing the cosine similarity cut-off

We used the approach implemented by the SBERT authors for choosing a cosine-similarity threshold [3]. This algorithm picks the threshold that makes the most accurate prediction for classifying both similar sentences and dissimilar sentence pairs on the test dataset.

3.6 Clustering Algorithm

We chose the agglomerative clustering algorithm for our study for the grouping task, as it does not require deciding the number of clusters beforehand. This algorithm initially assigns each sentence (embedding vector) to its cluster and afterward repeatedly merges pairs of clusters until all the clusters merge into a single cluster and form an agglomerative tree.

3.7 Evaluation Metrics

For comparing the sentence-embedding generator model's performance on the Sentence-Embedding-Test dataset, we used accuracy as a metric. For measuring the clustering performance, we used the silhouette coefficient.

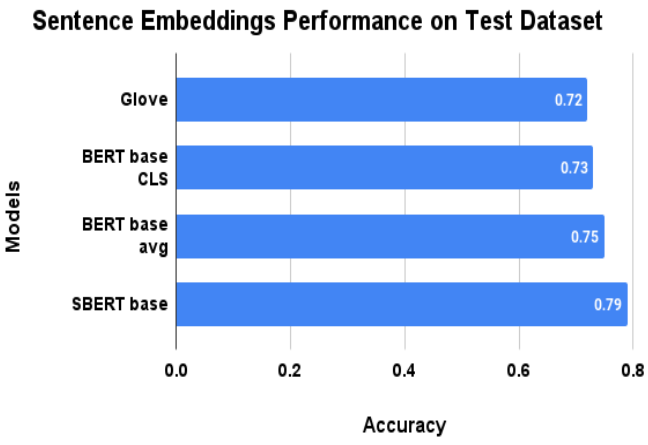


Figure 1: Comparison of Sentence Embedding Approaches using Accuracy Score on the Sentence-Embedding-Test Dataset

4. RESULTS AND DISCUSSION

This section presents the experimental results and discusses the findings of the research questions mentioned in section 1.

RQ1: Which pre-trained feature extraction methods for context and semantically related sentences work best?
We used accuracy as a metric to compare the performance of different sentence feature extraction methods on the Sentence-Embedding-Test dataset.

- Sentence pairs were identified as agreement or disagreement from GloVe feature embeddings with an accuracy score of 0.72. The classification accuracy score was 0.73 using feature embeddings from the BERT model's [CLS] token and 0.75 using the mean-pooling (average) method of BERT word embeddings. The base SBERT model had an accuracy score of 0.79. (Figure: 1)
- Considering the accuracy scores, base SBERT feature extraction performed best.

RQ2: Can we fine-tune and improve the pre-trained SBERT model's sentence-feature extraction for our context-dependent review text using a semi-supervised approach?

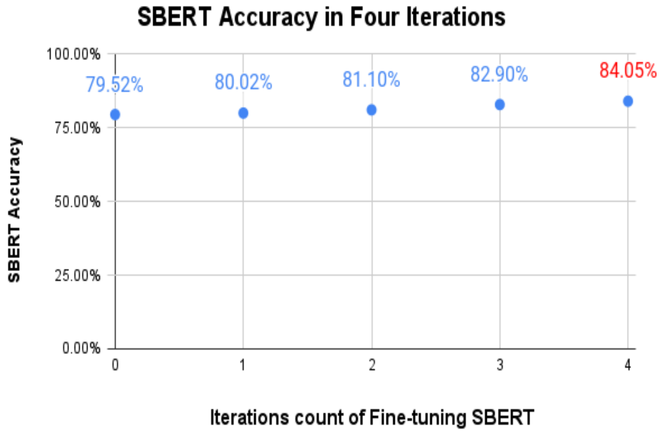


Figure 2: Fine-tuning of SBERT increased accuracy for identifying sentence similarity or difference after each iteration

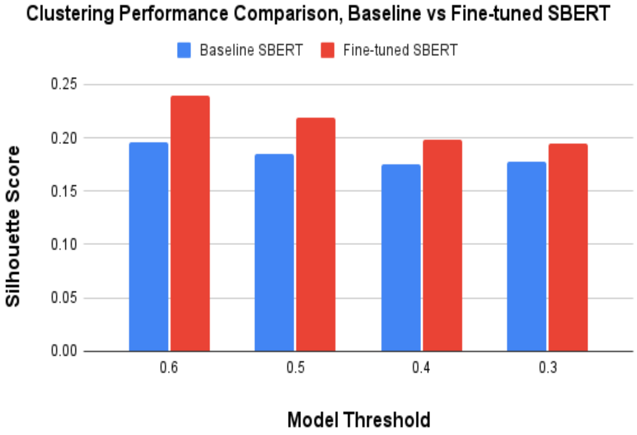


Figure 3: The clustering performance comparison using Silhouette Score with different thresholds for Agglomerative Hierarchical Clustering shows that Fine-tuned SBERT using Active Learning outperformed at every threshold point

Based on the accuracy scores for classifying the review-comment pair as agreeing or disagreeing, we picked the baseline Pre-trained SBERT model for further fine-tuning. We used the Fine-Tuning dataset and active-learning approach to fine-tune the SBERT model. We compared the model's performance using accuracy scores on the test Sentence-Embedding-Test dataset. Fine-tuning using active learning is an iterative method, so we continued the fine-tuning for four iterations. The result showed that the fine-tuned SBERT model improved accuracy after every iteration (Figure 2).

RQ3: Does improving the sentence feature extraction method improve clustering performance?
To test clustering performance, we used the Clustering-Test dataset. For each rubric item, this dataset has 2–5 review comments for each piece of work. We measure the clustering performance of both the baseline SBERT and fine-tuned SBERT using the silhouette score. For every clustering threshold we experimented with, the silhouette score for fine-tuned SBERT was higher than for the baseline SBERT model (Figure: 3).

5. CONCLUSION

In this study, we aim to identify disagreements in peer assessors' formative feedback by implementing a clustering algorithm. Our hypothesis is that reviews expressing similar feedback on a piece of work will be contextually and semantically similar, and that a clustering algorithm will be able to identify the similarity and put the similar feedback in a single group or cluster. On the other hand, feedback that expresses different opinions will be identified by the clustering algorithm and should be separated from other feedback. We showed that the performance of the clustering algorithm depends on the quality of the feature vectors that express the reviewers' natural language as machine-readable numbers. We have experimented with several baseline feature-vector extraction methods and fine-tuned SBERT sentence-embedding methods to compare quality. We carefully constructed the datasets for our experiments from reviews in a course that implemented the peer-assessment process. For fine-tuning the SBERT model, we implemented a semi-supervised active learning approach using uncertainty sampling and expert annotation. Our study showed that the fine-tuned SBERT sentence-embedding model outperformed the baseline SBERT model on our test dataset. Finally, we used the base-case model and the fine-tuned model's sentence embedding with the agglomerative clustering algorithm. We experimented with different thresholds and compared our results using silhouette scores. The silhouette score and empirical study of the clusters formed by the fine-tuned model show that the clustering algorithm can identify disagreements in peer-reviewers' formative feedback.

The key findings of this study are that base SBERT model outperforms other feature-extraction methods like Glove and BERT on the task of finding semantic review similarities on a peer-review dataset containing a high amount of software jargon. Also, we show that fine-tuning SBERT on this context-specific data further improves the model accuracy. We also show that fine-tuning improves the clustering done on the peer-review data to find disagreement in the review comments.

Since disagreement among reviewers can confuse students and lead them to question the review process, finding disagreements can help resolve the confusion by engaging reviewers in discussion and suggesting that the instructor intervene. In the future, we intend to extend this work to implement a recommendation system for reviewers to consider revising their feedback based on key points that other reviewers have identified.

6. REFERENCES

[1] J. Cambre, S. Klemmer, and C. Kulkarni. Juxtapeer: Comparative peer review yields higher quality feedback and promotes deeper reflection. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2018.

[2] Y. D. Çevik. Assessor or assessee? investigating the

differential effects of online peer assessment roles in the development of students' problem-solving skills. *Computers in Human Behavior*, 52:250–258, 2015.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] E. F. Gehringer. A survey of methods for improving review quality. In *International Conference on Web-Based Learning*, pages 92–97. Springer, 2014.

[5] M. H. Graner. Revision workshops: An alternative to peer editing groups. *The English Journal*, 76(3):40–45, 1987.

[6] R. Guan, H. Zhang, Y. Liang, F. Giunchiglia, L. Huang, and X. Feng. Deep feature-based text clustering and its explanation. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[7] S. Hiray and V. Duppada. Agree to disagree: Improving disagreement detection with dual grus. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pages 147–152. IEEE, 2017.

[8] S. Jinarat, B. Manaskasemsak, and A. Rungsawang. Short text clustering based on word semantic graph with word embedding model. In *2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS)*, pages 1427–1432. IEEE, 2018.

[9] L. Li and V. Grion. The power of giving feedback and receiving feedback in peer assessment. *All Ireland Journal of Higher Education*, 11(2), 2019.

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[11] R. Rada et al. Collaborative hypermedia in a classroom setting. *Journal of Educational Multimedia and Hypermedia*, 3(1):21–36, 1994.

[12] M. P. Rashid, E. F. Gehringer, M. Young, D. Doshi, Q. Jia, and Y. Xiao. Peer assessment rubric analyzer: An nlp approach to analyzing rubric items for better peer-review. In *2021 19th International Conference on Information Technology Based Higher Education and Training (ITHET)*, pages 1–9. IEEE, 2021.

[13] M. P. Rashid, Y. Xiao, and E. F. Gehringer. Going beyond" good job": Analyzing helpful feedback from the student's perspective. *International Educational Data Mining Society*, 2022.

[14] P. M. Sadler and E. Good. The impact of self-and peer-grading on student learning. *Educational assessment*, 11(1):1–31, 2006.

[15] K. Topping. Peer assessment between students in colleges and universities. *Review of educational Research*, 68(3):249–276, 1998.

Sequencing Educational Content Using Diversity Aware Bandits

Colton Botta
University of Edinburgh
cgb45@cam.ac.uk

Avi Segal
Ben-Gurion University
avise@post.bgu.ac.il

Kobi Gal
Ben-Gurion University
University of Edinburgh
kobig@bgu.ac.il

ABSTRACT

One important function of e-learning systems is to sequence learning material for students. E-learning systems use data, such as demographics, past performance, preferences, skillset, etc. to construct an accurate model of each student so that the sequencing of educational content can be personalized. Some of these student features are "shallow" traits which seldom change (e.g. age, race, gender) while others are "deep" traits that are more volatile (e.g. performance, goals, interests). In this work, we explore how reasoning about this diversity of student features can enhance the sequencing of educational content in an e-learning environment. By modeling the sequencing process as a Reinforcement Learning (RL) problem, we introduce Diversity Aware Bandit for Sequencing Educational Content (DABSEC), a novel contextual multi-armed bandit algorithm that leverages the dynamics within user features to cluster similar users together when making sequencing recommendations.

Keywords

Reinforcement Learning, Contextual Multi-Armed Bandit, Educational Sequencing

1. INTRODUCTION

Advancements in Artificial Intelligence (AI) have resulted in vastly improved models of student learning [4, 11, 14, 19]. Algorithms that use these models rely on data that describes students' online interactions, as well as their demographic information, previous academic performance, success on diagnostic questions, etc. All of this data can be collectively referred to as the *context* of the student, and it is within such contexts that algorithms operate in order to decipher how students are learning and how to best aid them. How these varying contextual features collectively model the complexities of human beings is of particular interest in this work, an idea we refer to as *human contextual diversity*. The advancement of e-learning technologies have brought together students of varied backgrounds and learning behaviors into single platforms, and reasoning about the diversity this creates when sequencing educational content is critical. We hypothesize that combining insights from social science about diversity can enrich educational models of students' behavior and improve the performance of educational sequencing algorithms. This work addresses the following questions: How can a machine detect human contextual diversity in educational data? Can we leverage the diverse and dynamic nature of this human data to improve how we sequence educational content to students?

To address these questions, we present a novel reinforcement learning algorithm, Diversity Aware Bandit for Sequencing Educational Content (DABSEC). DABSEC is a "diversity aware"[20] Contextual Multi-Armed Bandit (CMAB) algorithm with three main steps: calculate the dynamics of the underlying human contextual diversity in a group, form clusters of users with similar feature dynamics, and utilize these clusters and past student performance to sequence learning content to students. We compare the performance of DABSEC against LOCB [1], a state-of-the-art contextual bandit algorithm, as a baseline on two public educational datasets. Our results show that DABSEC achieves a higher average reward than LOCB on each dataset when predicting students' responses to questions.

2. BACKGROUND

We give an overview of CMAB algorithms and diversity.

2.1 Contextual Multi-Armed Bandits

Prior work has established that Bandit Algorithms, and RL in general, are effective solutions to educational sequencing[6]. One type of Bandit Algorithm, the Contextual Multi-Armed Bandit (CMAB) is a simplification of the full RL-problem and an extension of the Multi-Armed Bandit (MAB) problem where, at each timestep, the agent is presented with a list of arms (possible actions). Additionally, and unlike the original MAB setup, the agent is also presented with context (additional data) about the environment. The goal of the agent is to select a single arm, resulting in that action being performed. The agent then receives a reward for that arm only. Over time, the agent learns the underlying reward distribution of each arm and how that distribution is influenced by the context, and endeavors to maximize the total reward received over time [22].

CMABs have been used to sequence instructional material to students to increase overall learning [23, 15, 12], recom-

C. Botta, A. Segal, and K. Gal. Sequencing educational content using diversity aware bandits. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 502–508, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115731

mend news articles to readers [16], recommend the position of e-commerce items to maximize the chance a user interacts with them online [10], and many other use cases [3]. One recent work introduced the Local Clustering in Bandits (LOCB) algorithm [1] which implemented a "soft" clustering approach, by which users are clustered together if their preferences are within a certain threshold of each other. In this work, we use CMAB to select questions that students are most likely to get correct based upon their past question answering sequence.

2.2 Diversity

The existence of differences between humans in a group is one notion of diversity [2], with these differences often falling into two distinct categories: surface-level differences and deep-level differences [9]. Surface-level differences include, for example, age, sex, ethnicity, and race and are generally defined by their low-dynamics and ability to be observed immediately [13]. Deep-level differences, on the other hand, may include skills, values, preferences, and desires. These are more volatile and can only be observed through prolonged interaction between people [9]. For our purposes, we define *surface-level diversity* and *deep-level diversity* as differences between humans with respect to their surface-level and deep-level differences, respectively. One example of the importance of this classification is highlighted by the WeNet project, which places human diversity at the center of a new machine mediated paradigm of social interactions [2].

3. DABSEC

This section details the Diversity Aware Bandit for Sequencing Educational Content (DABSEC) algorithm.

3.1 Problem Definition

Assume $N = \{1, ..., n\}$ representing a set of n total users and $T = 1, ..., t$ representing a sequence of timesteps. At a timestep, t, a user, i_t, is drawn such that $i_t \in N$. Alongside i_t, the agent receives the context, $C_t = \{c_{1,t}, c_{2,t}, ..., c_{k,t}\}$ with one context vector for each of k arms and each context vector having dimension d such that $c_{k,t} \in \mathbb{R}^d$. The agent chooses one context vector, $c_{k,t}$, associated with arm $x_{k,t}$, to recommend to i_t and receives reward r_t in return. We assume that each user is associated with an unknown bandit parameter $\theta_{i,t}$ that describes how i_t interacts with the environment and can be thought of as a representation of how user i_t behaves [1]. As in previous bandit settings [16, 1, 7], the goal is to minimize the total regret, R_T given by:

$$R_T = \sum_{t=1}^{T} [\theta_{i,t}^{\intercal}(argmax_{c_{k,t} \in C_t} \theta_{i,t}^{\intercal} c_{k,t}) - \theta_{i,t}^{\intercal} c_t] \qquad (1)$$

where, at each round, t, we compute the regret by taking the reward achieved from the best possible arm choice, $x_{k,t}$, and subtracting the reward achieved from the agent's chosen arm, x_t. We also assume that each user, i, has a set of features, F, of length q such that at any time, t, there exists $F_{i,t} = \{f_{i,1,t}, f_{i,2,t} ..., f_{i,q,t}\}$.

3.2 DABSEC Algorithm

The DABSEC algorithm has three main steps: calculate the underlying feature dynamics of all users over time, form clusters of users with similar feature dynamics, then utilize the clusters and past student performance to sequence learning content to students. DABSEC (Algorithm 1) is initialized with the number of clusters to maintain (s), the frequency with which to update the clusters ($T_{cluster}$), the frequency with which to update the user feature dynamics ($\mathcal{U}$), and an exploration parameter (α). Then, all users are initialized (Lines 2-4) and the algorithm begins iterating over all timesteps sequentially (Line 5). In each round, t, a user i_t is presented along with the set of context vectors C_t (Line 6). DABSEC begins without any user clusters. DABSEC first checks if there are any clusters (Line 7), and if there are none (length($\mathcal{G} \leq 0$)), then the arm with the highest upper confidence bound (UCB) is chosen. As is standard practice [16] in bandit algorithms, UCB is computed using the estimation of user i_t's unknown bandit parameter, $\hat{\theta}_{i,t}$ (Lines 14-16) where $A_{i,t-1}^{-1}$ is the covariance matrix and $b_{i,t-1}$ is a normalizing matrix for user i at timestep $t-1$ that are used to compute the ridge regression solution of the coefficients [16]. On the other hand, if a user clustering has been established (length($\mathcal{G} > 0$)), then the cluster holding user i_t is set as $g_{s,t}$ (Line 8) and DABSEC calculates $\hat{\theta}_{g_{s,t}}$, which represents the unknown bandit parameter for the entire cluster (Line 9).

Finally, to choose an arm, we compare the UCB using the user's unknown bandit parameter, $\hat{\theta}_{i,t}$ to the UCB using the average unknown bandit parameter of all users in cluster $g_{s,t}$, $\hat{\theta}_{g_{s,t}}$ (Lines 10-12). The maximum of these two UCB values is selected (Line 13). The reasoning behind this is that previous work has established that clustering users by unknown bandit parameter is an effective strategy for identifying users who behave similarly in a task, thus resulting in a collaborative filtering effect [8, 7, 17, 18, 1]. In datasets where changes in user features are not available or considered, these past works still represent the state of the art in clustering bandit algorithms. Our approach, by comparison, is to gain an advantage in datasets where user feature dynamics are available and changing. In these cases, we expect the collective bandit parameter of the cluster where user i_t resides, $\hat{\theta}_{g_{s,t}}$, to estimate expected behavior better than $\hat{\theta}_{i,t}$.

With an arm chosen and pulled, we observe the reward, r_t, then update user parameters and cluster parameters for the cluster that user i_t resides in (Lines 17-22). Then, any user features, $F_{i,t}$ are updated (Lines 23-24). This step will be tailored to the specific implementation and dataset, as the number, type, and sophistication of the user features will be entirely dependent on the problem definition and setup. The count for how many times user i_t has been considered is also updated (Line 25). Finally, the most up to date clusters, $\mathcal{G}_t$, are calculated and returned by the CLUSTER function (Line 26 - see Algorithm 2), which ends round t.

3.3 Clustering by User Feature Dynamics

The second component of DABSEC is clustering users based upon the similarity of their feature dynamics. The CLUSTER algorithm (Algorithm 2) assumes that each user has a set of features, F, of length q such that at any time, t, there exists $F_{i,t} = \{f_{i,1,t}, f_{i,2,t} ..., f_{i,q,t}\}$. The values of each individual user feature, $f_{i,q,t}$ may change over time, which can be tracked to cluster users based upon the similarity of their feature dynamics. To do this, one can observe the value of a feature at some initial timestep, then again at a later timestep, and calculate the absolute value of the difference between them. More formally, at some initial timestep, $T_{initial}$, we store the values of all features for a given user, i_t: $F_{i_t,T_{initial}}$. We also initialize a set Y_t that contains one value for each user such that $Y_t = \{y_{1,t}, y_{2,t}...y_{i,t}\}$ and $y_{i,t}$ represents the number of times that the agent has made a recommendation to user i_t. Thus, each time user i_t is selected by the algorithm, we can update $F_{i,t}$ based upon the observed user features at timestep t, and increment $y_{i,t}$ by 1. Once the agent has made a recommendation to a user $\mathcal{U}$ times, say at time T_{final}, such that $y_{i,t} = \mathcal{U}$, the feature dynamics for user i, δ_i, can be computed based upon how the features have changed between $T_{initial}$ and T_{final} (Algorithm 2 Line 2). The differences are summed over time to compute δ_i, and $\mathcal{U}$ is a hyperparameter that controls how often user feature dynamics are updated. After this calculation, $T_{initial}$ is set to T_{final} and $y_{i_t,t}$ is set to 0. The process repeats when $y_{i_t,t} = \mathcal{U}$ until all timesteps are complete.

By performing this operation for every user, we constantly have access to δ_i which represents the current dynamics of user i's features. We use the similarity between user's δ values to cluster them together, rather than $\theta_{i,t}$ as done in previous works [8, 7, 17, 18, 1]. To that end, we assume that there exists a set of clusters $\mathcal{G}$ of length s such that $\mathcal{G}_t = \{g_{1,t}, g_{2,t}...g_{s,t}\}$. For simplicity, we assume that each user must appear in exactly one cluster and all users are split evenly amongst the clusters. This results in each cluster containing $\frac{n}{s}$ users. See Algorithm 2 for the full clustering pseudocode.

DABSEC updates clusters after a period of timesteps have passed $T_{cluster}$. This is because calculating the dynamics of the user features requires observing changes in those features over a period of time. To re-cluster after every timestep would not allow sufficient time to observe any true dynamics, so we update δ_i for each user after every $\mathcal{U}$ timesteps in which that user is selected.

Algorithm 1 DABSEC

```
Require: number of clusters to form s, cluster update frequency $T_{cluster}$, user feature dynamics update frequency $\mathcal{U}$, exploration parameter $\alpha$
$T_{initial} \leftarrow 0$
for each $i \in N$ do
    $A_{i,0} \leftarrow I, b_{i,0} \leftarrow 0$
    $y_i \leftarrow 0$
for $t \leftarrow 1, 2...T_{final}$ do
    receive $i_t \in N$ and obtain $C_t = \{c_{1,t}, c_{2,t}..., c_{k,t}\}$
    if length of $\mathcal{G} \geq 0$ then
        $g_{s,t} \leftarrow$ Cluster where $i_t$ resides at round t
        $\hat{\theta}_{g_{s,t}} \leftarrow \frac{1}{|g_{s,t-1}|} \sum_{j \in g_{s,t-1}} A_{j,t-1}^{-1} b_{j,t-1}$
        $x_{cluster} \leftarrow argmax_{c_{a,t} \in C_t} \hat{\theta}_{g_{s,t}}^{\intercal} c_{a,t} + CB_{r,g_{s,t}}$ where $CB_{r,g_{s,t}} \leftarrow \frac{1}{|g_{s,t-1}|} \sum_{j \in g_{s,t-1}} \alpha \sqrt{c_{a,t}^{\intercal} A_{j,t-1}^{-1} c_{a,t}}$
        $\hat{\theta}_{i_t t} \leftarrow A_{i,t-1}^{-1} b_{i,t-1}$
        $x_{user} \leftarrow argmax_{c_{a,t} \in C_t} \hat{\theta}_{i_t t}^{\intercal} c_{a,t} + CB_{r,i}$ where $CB_{r,i} \leftarrow \alpha \sqrt{c_{a,t}^{\intercal} A_{i,t-1}^{-1} c_{a,t}}$
        $x_t \leftarrow max(x_{cluster}, x_{user})$
    else
        $\hat{\theta}_{i_t t} \leftarrow A_{i,t-1}^{-1} b_{i,t-1}$
        $x_t \leftarrow argmax_{c_{a,t} \in C_t} \hat{\theta}_{i_t t}^{\intercal} c_{a,t} + CB_{r,i}$ where $CB_{r,i} \leftarrow \alpha \sqrt{c_{a,t}^{\intercal} A_{i,t-1}^{-1} c_{a,t}}$
    pull $x_t$ and observe reward $r_t$
    $A_{i,t} \leftarrow A_{i,t-1} + x_t x_t^{-1}$
    $b_{i,t} \leftarrow b_{i,t-1} + r_t x_t$
    if length of $\mathcal{G} \geq 0$ then
        $A_{g_{s,t},t} \leftarrow A_{g_{s,t},t-1} + x_t x_t^{-1}$
        $b_{g_{s,t},t} \leftarrow b_{g_{s,t},t-1} + r_t x_t$
    for $f_{i,q,t} \in F_{i,t}$ do
        update $f_{i,q,t}$ according to information gathered from problem setup and $r_t$
    $y_{i,t} \leftarrow y_{i,t} + 1$
    $\mathcal{G}_t \leftarrow CLUSTER(\mathcal{U}, Y, T_{cluster}, i_t)$
```

4. DABSEC ON EDUCATION DATA

In this section, we apply the DABSEC algorithm to two large-scale educational datasets: Eedi [24] and EdNet [5].

4.1 Eedi Dataset

Eedi[1] released a dataset that includes over 17 million interactions of students answering multiple choice questions. It was used for The NeurIPS 2020 Education Challenge [24] and contains two identically structured halves: Eedi1 and Eedi2. Each provides interaction logs of the student ID, question ID, student answer (range a-d), and the correct answer (range a-d). Every question has an associated list of features including a question ID, and a list of subject IDs (a list of IDs that correspond to mathematics concepts that are covered by the question). Every student has an associated list of features including gender, date of birth and a boolean indicator if the student is financially disadvantaged or not.

4.2 EdNet Dataset

The EdNet dataset[5] was the largest publicly-available education dataset when it was released in 2020. It contains over 131 million interactions from over 784,000 students who, over the course of two years, used the Santa[2] platform to study English for the Test of English for International Communication (TOEIC) exam. The dataset is organized in a 4-level, hierarchical style, and we consider the KT1 version for our analysis. The KT1 dataset is a collection of 784,309 CSV files, where each file contains the question answering logs of one student. Each line represents a question that the student answered, and includes the timestamp of the answer submission, a solving ID, the ID of the answered question, the student's answer (from a-d), and the amount of time spent answering the question. For each of the 13,169 questions in the dataset, the correct solution and the question tags are provided. These question tags are identical to the concept of subjects from the Eedi dataset described in section 4.1. We refer to the tags as subjects for consistency.

4.3 Experiments

In this section we describe an educational setting where an agent trained using DABSEC chooses personalized sequences of mathematics questions, based upon past student performance, that are likely to be answered correctly by the

[1] https://eedi.com

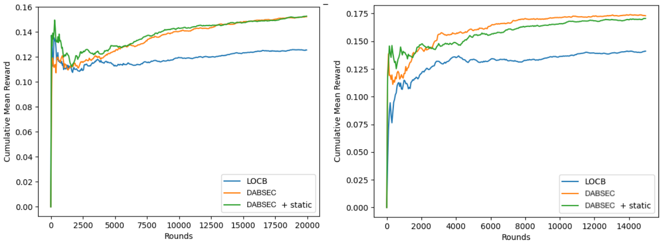
[2] https://www.aitutorsanta.com

Algorithm 2 *CLUSTER*

```
Require: user feature dynamics update frequency $\mathcal{U}$, user update counts Y, cluster update frequency $T_{cluster}$, user $i_t$
if $y_i == \mathcal{U}$ then
    $\delta_i = \sum_{q=1}^{Q}\{|F_{i,t} - F_{i,T_{initial}}|\}$
    $T_{initial} \leftarrow t$
    $y_i \leftarrow 0$
if t % $T_{cluster} == 0$ then
    $\delta_{sorted} \leftarrow$ sort $\delta$ in ascending order
    $\mathcal{G}_t \leftarrow$ split($\delta_{sorted}$,s) where split(x,y) splits x into $length(x)\%y$ groups each of size $\frac{length(x)}{y} + 1$ and the rest of size $\frac{length(x)}{y}$
return $\mathcal{G}_t$
```

student. We apply DABSEC to Eedi1, Eedi2 and EdNet, by first obtaining the full list of unique questions that each student answered, along with the subject categories, the student answer, and correct answer for each question. At each round where user i_t is selected, we randomly sample 10 questions that student i_t has answered. Because we are interested in building an agent that can identify questions that each student should be able to answer correctly, we follow a recent approach [1] of selecting 9 questions that the student answered incorrectly in the past, and 1 question that the student answered correctly in the past. The correct question is not revealed to the agent. Not all students in the dataset answered enough total questions to be considered in this experimental setup, so we selected a subset: for the Eedi datasets, we consider the 50 users with the most total questions answered. For the EdNet dataset, we sample 50 users who have answered over 1000 questions. Thus, during each round of DABSEC, the agent receives a user, i_t, a list of 10 random questions that i_t has answered in the past (9 incorrect, 1 correct) and a context vector that contains the student's past performance by subject. The agent then chooses 1 question that it believes i_t is mostly likely to answer correctly. The agent is given a reward of 1 if it correctly selects the 1 question that user i_t did answer correctly in the past, and a reward of 0 otherwise. To compare the performance across datasets and against the baseline, we calculate and report the cumulative average reward achieved over every sequence of 50 timesteps.

Using the above setup, we first applied the original LOCB algorithm to both datasets. The creators released an open-source implementation of LOCB[3] which we extended and adapted to operate on our datasets. After the base setup, the algorithm continually forms and updates clusters based on the similarity of student's unknown bandit parameter, θ, which is a proxy for student preferences and behavior as discussed in Section 3. At each timestep, LOCB computes the average θ of the current student's cluster and uses it to select the question that was most likely answered correctly. In the original work's main experiments, the authors conclude that setting the number of clusters to 20, gamma to 0.2 and delta to 0.1 would return good results on average, so we use these values for our LOCB implementation.



(a) Eedi1 Dataset

(b) Eedi2 Dataset

Figure 1: A comparison of the performance of DABSEC, DABSEC + static, and LOCB on both Eedi datasets based on cumulative average reward.

We then applied DABSEC to all datasets, with clusters being continually updated every $T_{cluster}$ timesteps based on the average bandit parameter, θ, of a user's cluster, where clusters are formed based on similarity of feature dynamics as discussed in Section 3. We set the following hyperparameters for both datasets: $T_{cluster} = 1000$, $\mathcal{U} = 10$, and $s = 3$, as these produced the best overall performance. Additional hyperparameter settings are described in Appendix A.

Finally, for the Eedi dataset only, we follow an identical setup as DABSEC described above with the addition of the static (low dynamic) student features: the age, gender, and if they are financially disadvantaged. We call this DABSEC + static. We do not apply DABSEC + static to the EdNet dataset because there are no demographic features.

5. RESULTS AND ANALYSIS

We compare the performance of DABSEC, DABSEC + static, and LOCB on all datasets, and describe DABSEC's potential educational applications.

5.1 Results

As shown in Figure 1a, both of the DABSEC variations outperform the LOCB baseline by nearly 30% with respect to cumulative mean reward obtained over time on the Eedi1 dataset. Neither DABSEC variation seems to outperform the other. Looking at Figure 1b, we see that both DABSEC variations again outperform the LOCB baseline on the Eedi2 dataset - this time by about 25%. In this dataset, DABSEC slightly outperforms DABSEC + static but the gap is nearly closed by the time we reach the end of the rounds. Finally, in Figure 2, DABSEC outperforms the LOCB baseline by over 30% on the EdNet dataset.

Our experimental results confirm that DABSEC achieves better performance than LOCB on the Eedi1, Eedi2, and EdNet datasets. We found evidence that identifying and extracting feature dynamics can improve RL algorithm performance, and that clustering users based on their feature dynamics, rather than estimated user preferences alone, is a good starting towards improving clustering algorithms based on human diversity. We argue that the reason for this improvement is that identifying the highly dynamic features allows DABSEC to search the space of context-reward associations more completely and more quickly, thus leading to better reward. The low dynamic, static features, on the other hand, either exclude part of the search space or explore

[3] https://github.com/banyikun/LOCB

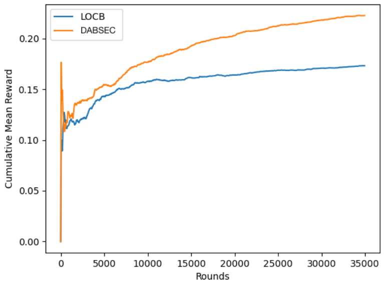


Figure 2: A comparison of the performance of DABSEC and LOCB on the EdNet dataset based on cumulative average reward. We ran 35,000 rounds until seeing evidence of stabilization.

it more slowly than DABSEC is capable of learning, leading to lower reward over the same timespan. This theory requires further testing, but the results of applying DABSEC to real data are promising, and further research into augmenting our clustering approach is planned for the future.

5.2 Implications for Education

We believe that a diversity aware approach to RL has high potential in the education domain. Due to the amount of individual behavioral data, one of the dominant use cases of RL and Bandit algorithms is e-learning systems, where students answer questions while the system attempts to observe, understand, and improve student knowledge based on the responses [21]. This is an ideal environment where user features are highly dynamic, as student performance across subjects changes with each question answered. This is a phenomenon we saw in our experiments in Section 4 and were able to exploit to boost performance. We believe that there is a potential for algorithms like DABSEC to further improve e-learning technology.

6. CONCLUSION

In this work, we designed, implemented, and tested DABSEC, a diversity aware RL algorithm that uses feature dynamics as a proxy for underlying human-contextual diversity, then clusters users based on this metric. We hypothesized that this technique could improve RL algorithms that operate in environments where user data is highly dynamic, and this proved true when applying DABSEC to two large-scale educational datasets. DABSEC outperforms the LOCB baseline by approximately 30% based on cumulative mean reward earned over time, and we believe that extensions to DABSEC can make it an ideal tool for building more performant e-learning applications.

6.1 Limitations

Our approach is an initial attempt to develop a diversity aware RL approach that leverages the dynamics of human data over time. One major drawback is that if a dataset is mostly comprised of features with low dynamics, the user feature dynamics would always be calculated as near zero and the clusters would be far less informative. Similarly, our assumption that user's could only be in one cluster may fall short of fully capturing the most available data on every student, as LOCB found by letting user's reside in multiple clusters simultaneously [1]. Similarly, by requiring all clusters to include the same number of users, we may not be forming the ideal clusters - for example, if the cluster size dictates that each cluster should have 10 users, but there are 3 users that are extreme outliers, then these 3 might benefit from residing in their own cluster. Additionally, in our definition of diversity, we assume that user features that remain constant are likely surface-level, whereas more dynamic features are likely deep-level. Of course, this may not hold in all situations; some people's goals, personalities, and values may never change, despite being classified as traits of deep-level diversity. For the sake of this work, we make this assumption based upon past sociology research [9, 13], but acknowledge that it may not hold in all implementation use cases. Finally, we followed the experimental approach that LOCB[1] used by randomly selecting the data at each round - we picked the student randomly, then randomly chose 9 questions that the student got incorrect and 1 that the student got correct to serve as the arms. This assumes knowledge of the entire dataset at the beginning, which would not be the case in real-time e-learning systems which consider student interactions as they occur.

6.2 Future Work

Further research should be conducted to improve upon our initial findings. First, there is an opportunity to improve the clustering algorithm to account for additional data about the user. For example, users could be clustered using a combination of overall feature dynamics and the preferences of users, represented by their unknown bandit parameter θ. This technique may boost performance by clustering users based upon both their preferences and how those preferences are changing over time. Second, this work included running DABSEC on two real-world educational datasets, but deploying DABSEC in the wild would offer further insight into the usefulness of diversity-aware RL. We would like to deploy DABSEC in a live e-learning platform so that it can sequence learning content to students in real-time. Finally, given that incorporating human data and diversity within algorithms needs to be handled with care, an exciting extension of this work would be to consider if diversity-aware algorithms have any implications on algorithmic fairness. For instance, investigating whether or not algorithmic fairness is more easily achieved with a diversity-aware algorithm, or if diversity-aware algorithms are more or less transparent than traditional algorithms are both important research areas to explore.

7. ACKNOWLEDGEMENTS

This work was supported in part by the European Union Horizon 2020 WeNet research and innovation program under grant agreement No 823783.

8. REFERENCES

[1] Y. Ban and J. He. Local clustering in contextual multi-armed bandits. In *Proceedings of the Web Conference 2021*, pages 2335–2346, 2021.

[2] I. Bison, M. Bidoglia, M. Busso, R. C. Abente, M. Cvajner, M. D. R. Britez, G. Gaskell, G. Sciortino, S. Stares, et al. D1. 3 final model of diversity: Findings from the pre-pilots study. 2021.

[3] D. Bouneffouf, I. Rish, and C. Aggarwal. Survey on applications of multi-armed and contextual bandits. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2020.

[4] B. Choffin, F. Popineau, Y. Bourda, and J.-J. Vie. Das3h: modeling student learning and forgetting for optimally scheduling distributed practice of skills. *arXiv preprint arXiv:1905.06873*, 2019.

[5] Y. Choi, Y. Lee, D. Shin, J. Cho, S. Park, S. Lee, J. Baek, C. Bae, B. Kim, and J. Heo. Ednet: A large-scale hierarchical dataset in education. In *International Conference on Artificial Intelligence in Education*, pages 69–73. Springer, 2020.

[6] S. Doroudi, V. Aleven, and E. Brunskill. Where's the reward? a review of reinforcement learning for instructional sequencing. *International Journal of Artificial Intelligence in Education*, 29:568–620, 2019.

[7] C. Gentile, S. Li, P. Kar, A. Karatzoglou, G. Zappella, and E. Etrue. On context-dependent clustering of bandits. In *International Conference on Machine Learning*, pages 1253–1262. PMLR, 2017.

[8] C. Gentile, S. Li, and G. Zappella. Online clustering of bandits. In *International Conference on Machine Learning*, pages 757–765. PMLR, 2014.

[9] D. A. Harrison, K. H. Price, and M. P. Bell. Beyond relational demography: Time and the effects of surface-and deep-level diversity on work group cohesion. *Academy of management journal*, 41(1):96–107, 1998.

[10] X. He, B. An, Y. Li, H. Chen, Q. Guo, X. Li, and Z. Wang. Contextual user browsing bandits for large-scale online mobile recommendation. In *Fourteenth ACM Conference on Recommender Systems*, pages 63–72, 2020.

[11] J. He-Yueya and A. Singla. Quizzing policy using reinforcement learning for inferring the student knowledge state. *International Educational Data Mining Society*, 2021.

[12] W. Intayoad, C. Kamyod, and P. Temdee. Reinforcement learning based on contextual bandits for personalized online learning recommendation systems. *Wireless Personal Communications*, 115(4):2917–2932, 2020.

[13] S. E. Jackson, V. K. Stone, and E. B. Alvarez. Socialization amidst diversity-the impact of demographics on work team oldtimers and newcomers. *Research in organizational behavior*, 15:45–109, 1992.

[14] K. R. Koedinger, J. C. Stamper, E. A. McLaughlin, and T. Nixon. Using data-driven discovery of better student models to improve student learning. In *International conference on artificial intelligence in education*, pages 421–430. Springer, 2013.

[15] A. S. Lan and R. G. Baraniuk. A contextual bandits framework for personalized learning action selection. In *EDM*, pages 424–429, 2016.

[16] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

[17] S. Li, W. Chen, and K.-S. Leung. Improved algorithm on online clustering of bandits. *arXiv preprint arXiv:1902.09162*, 2019.

[18] S. Li, A. Karatzoglou, and C. Gentile. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 539–548, 2016.

[19] H. Nakagawa, Y. Iwasawa, and Y. Matsuo. Graph-based knowledge tracing: modeling student proficiency using graph neural network. In *2019 IEEE/WIC/ACM International Conference On Web Intelligence (WI)*, pages 156–163. IEEE, 2019.

[20] L. Schelenz, I. Bison, M. Busso, A. De Götzen, D. Gatica-Perez, F. Giunchiglia, L. Meegahapola, and S. Ruiz-Correa. The theory, practice, and ethical challenges of designing a diversity-aware platform for social relations. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 905–915, 2021.

[21] A. Singla, A. N. Rafferty, G. Radanovic, and N. T. Heffernan. Reinforcement learning for education: Opportunities and challenges. *arXiv preprint arXiv:2107.08828*, 2021.

[22] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[23] C. Tekin, J. Braun, and M. van der Schaar. etutor: Online learning for personalized education. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5545–5549. IEEE, 2015.

[24] Z. Wang, A. Lamb, E. Saveliev, P. Cameron, Y. Zaykov, J. M. Hernández-Lobato, R. E. Turner, R. G. Baraniuk, C. Barton, S. P. Jones, et al. Instructions and guide for diagnostic questions: The neurips 2020 education challenge. *arXiv preprint arXiv:2007.12061*, 2020.

APPENDIX

A. HYPERPARAMETER VARIATIONS

Using the DABSEC algorithm on the EdNet dataset, we also explored a few variations of the hyperparameters: the number of clusters to sort users into, s, the user feature dynamics update frequency, $\mathcal{U}$, and the cluster update frequency, $T_{cluster}$. Like before, we measure the performance based on cumulative mean reward achieved over time. Figure 3a shows the effect of changing the frequency with which the user feature dynamics are updated ($\mathcal{U}$). We held the number of clusters constant at 3 and the cluster update frequency constant at 1000. We set $\mathcal{U}$ as 5, 10, 50, and 100, which represent how many questions need to be answered by a user before we recalculate their current feature dynamics. We can see that the performance of DABSEC is not effected much by changing $\mathcal{U}$, though the best performing variation updated a user's feature dynamics after every 100 questions answered by that user. This makes sense, because a larger $\mathcal{U}$

forces a larger amount of questions to be answered between feature dynamics calculations, meaning that there will be far more data to consider than when $\mathcal{U}$ is smaller. However, the difference in performance is not very significant.

Figure 3b shows the effect of changing the frequency with which the actual global clusters are updated ($T_{cluster}$). We held the number of clusters constant at 3 and the user feature dynamics update frequency constant at 10. We set $T_{cluster}$ as 500, 1000, 2000, and 5000, which represent how many rounds occur between every instance of reclustering. We can see that the performance of DABSEC is not effected much by changing $T_{cluster}$, though the worst performing variation updated clusters every 500 rounds. This makes sense, because a smaller $T_{cluster}$ would not be considering as much data when forming new clusters, which may result in clusters that are less indicative of true similarities between users. It would make sense that a higher $T_{cluster}$ would result in more data being considered by the clustering algorithm, thus resulting in better clusters and a better performing algorithm. However, the difference in performance is not very significant.

Finally, Figure 4 shows the effects of changing the number of clusters that users are placed into. DABSEC achieves better performance when the number of clusters is smaller (3), with performance incrementally worsening as the number of clusters increases to 5, 10, and 15. This is in line with our expectations, as we are only using 50 total users which makes the size of the clusters quite small as the number of clusters increases. In the future, running these experiments with more total users would be interesting.

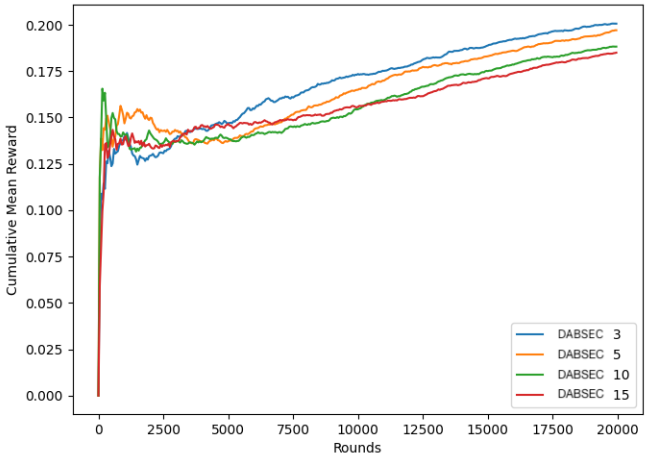
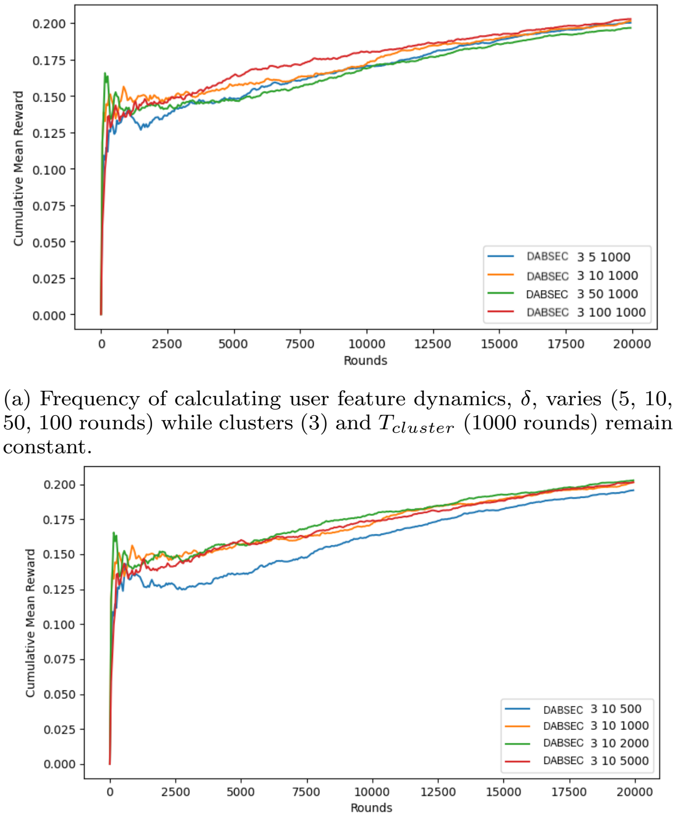


Figure 4: Cluster size varies (3, 5, 10, 15) while $T_{cluster}$ (1000) and feature dynamics update frequency (10 rounds) remains constant.



(a) Frequency of calculating user feature dynamics, δ, varies (5, 10, 50, 100 rounds) while clusters (3) and $T_{cluster}$ (1000 rounds) remain constant.

(b) Frequency of calculating the global clusters, $T_{cluster}$, varies (500, 1000, 2000, 5000 rounds) while clusters (3) and feature dynamics update frequency (10 rounds) remain constant.

Figure 3: Hyperparameter variations using DABSEC on the EdNet dataset.

A comparative analysis of the cognitive levels of Science and Mathematics secondary school board examination questions in India

Anirban Roy Chowdhury
IISER Pune
anirban.chowdhury@students.iiserpune.ac.in

Nandagopal K S
IISER Pune
nandagopal.ks@students.iiserpune.ac.in

Vijay Prakash
IIT Bombay
prakashvijay649@gmail.com

Syaamantak Das
IIT Bombay
syaamantak.das@iitb.ac.in

ABSTRACT

This research study investigates the cognitive levels of the questions used for assessment and evaluation purpose in three national school board secondary school leaving examinations in Mathematics and Science subjects, conducted in Indian schools from 2011 to 2020. The research used Bloom's Taxonomy to identify the cognitive levels of 3071 board examination questions. The study addresses the gap in the current literature on the non-availability of a comparative study of national school board assessment practices in India. The study provides a comparative analysis of three English medium Indian school boards (ICSE, CBSE, and NIOS) by analyzing what areas/topics the board exams test and which ones they ignore. Based on this analysis, certain trends were analysed which are present in boards/subjects in the 10th standard national level exams in India. This study will help school education stakeholders to get an insight into the cognitive level trend/patterns of assessment questions.

Keywords

Cognitive level, Assessment Questions, School Examination, Indian School Boards

1. INTRODUCTION

A learning process can be segmented into three sections [7]. First, the learning goals, which are also referred to as instructional objectives / Learning Objectives (LO). They specify the learning outcomes that mention the skills and knowledge which is to be imparted to the learner. Second, the selection of learning materials that meet the requirement to achieve the learning goals. Third, assessment questions that determine whether the learner is learning the necessary skills and knowledge to solve the problem. In the educational domain, the Learning Objective refers to the statement(s) of a course curriculum, specifically describing the skills and knowledge that the student must gain after the course is completed. If complemented by a similar cognitive level of instructional techniques and tests, the learning outcomes help the teachers determine whether the students are achieving the expected skills and knowledge.

The Learning Objective also determines the performance results by determining the conditions under which performance will occur. It also defines the requirements (specific skills, competencies, and attitudes) that the learners will follow. "Learning outcomes are precise statements of what faculty expects students to know and to be able to do in some measurable way as a result of completing a program, course, unit, or lesson" as stated in [3]. It was also observed that "in addition to guiding, teaching, learning, and assessment strategy, effective learning outcomes facilitate student orientation to the subject and communicate expectations" as reflected in this research [6]. For active learning to take place, 'there must be a constructive alignment of the curriculum, which should ensure that in an education program, the learning objectives, teaching and learning methodologies, and assessment techniques should complement each other'.

Student's ability to think for activities during instruction and examinations are essential for improving their intellectual abilities, performance execution, and professional growth. As a result, exams should be designed to encourage students to express their thoughts on the exam questions, develop creative answers, and connect the exam answers to their own experiences and real-life situations [5]. Also, good questions not only promote effective learning and assessment, but they must also be consistent with curriculum and instruction, as assessment has a significant impact on both learning and teaching. The phenomenon is known as the 'washback effect', and it refers to how testing affects teaching and learning [17].

As a result, writing high-quality exams that include both higher-level questions (HLQs) and lower-level questions (LLQs) is critical in assisting students in achieving the desired learning outcomes and evaluating their level of proficiency in a specific course. The HLQs help students dig deeper into the learning materials while also encouraging critical thinking

A. R. Chowdhury, N. K. S, V. Prakash, and S. Das. A comparative analysis of the cognitive levels of science and mathematics secondary school board examination questions in india. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 509–514, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115778

and creativity. A work by [2] stated that, schools must emphasise higher-order skills in order to develop critical thinking.

Therefore, the present study seeks to examine and compare to what extent the questions of the tenth grade school board examinations in Mathematics and Science subjects prepared by three national level Indian school boards include both higher and lower-order thinking levels. In the first place, it is essential to assess how well students master the information of the educational materials within the six levels of the Bloom's Taxonomy. Additionally, it is essential to analyze whether the exam questions of the given examinations in the Mathematics and Science subjects are based on both higher and lower-order thinking levels.

2. LITERATURE REVIEW

2.1 Mapping Bloom's Taxonomy Cognitive Levels to Thinking orders

Bloom developed a taxonomy in this context that is used to develop assessments that take into account each of the six levels of hierarchy in the cognitive domain [14]. Knowledge (recalling details), comprehension (description in someone else's words), and application (using existing knowledge to produce results) are examples of lower-order thinking domains. Higher-order thinking domains include analysis (discovering connections between facts and concepts), synthesis (creating new original work), and evaluation (judging and demonstrating one's position) [12]. Researchers used Bloom's Taxonomy's two cognitive categories to analyse and determine the levels of questions asked in exams, and they established two types: lower-level questions (LLQs) and higher-level questions (HLQs) [16]. The LLQs are designed to test students' recall of fundamental and universal concepts and processes. The HLQs, on the other hand, are more advanced and difficult because they require students to engage in deeper and analytical thinking processes.

2.2 School Board Examination System in India

The work by [8] gives a comprehensive overview of Indian school education system. The research article extensively covers the Indian school education system, which is one of the largest education system in the world. With both public and private schools, the Indian school system can be divided into four main categories – pre primary (consisting of pre-school, lower and higher kindergarten), primary school (standard one to five), middle school (standard six to eight), secondary school (standard nine and ten) and high school (standard eleven and twelve / or pre university standard). The public schools are majorly either central government schools (such as Kendriya vidyalayas, navodaya vidyalayas, Sainik schools etc.) or state government schools of respective states. The private schools are usually run by individuals, trusts or societies and may or may not receive fund from the government. Apart from these two major categories, some other semi government type schools run by local government bodies also exist (e.g. Municipality schools). The central government schools are usually affiliated to the Central Board for Secondary Education (CBSE) supervised by the National Council of Educational Research and Training (NCERT) under the Ministry of Education. The Council of Indian School Certificate Examinations (CISCE) is a semi-private, non governmental education board in India. It conducts the Indian Certificate for Secondary Education (ICSE) examination (for tenth standard) and Indian School Certificate (ISC) examination (for twelfth standard) in India. These two are the major all India based school examination boards. Apart from the all India based school examination boards, the state government affiliated school examination boards constitute a major part of Indian school examination system. All Indian school boards for the sake of collaborating and exchange of information with each other forms an umbrella body called Council of Boards of School Education in India (COBSE), a voluntary association of all the boards of school education in India. There are more than 50 members with associate members from Nepal, Mauritius, Bhutan, Pakistan and United Kingdom. Other than these Indian school boards, foreign school boards such as International Baccalaureate Organization (IBO) and Cambridge International Examinations (CIE) are emerging as newer school boards in urban areas. These schools boards offer global school level examinations all over the world and follows universal curriculum. It must be noted that all school boards in India are autonomous having their own syllabus, curriculum, method of assessment and evaluation.

An elaborate research work on the quality of school education in India has been done by Institute for Studies in Industrial Development as mentioned in [9] for Quality Council of India, New Delhi. Firstly, they have clearly defined the distinction between syllabus and curriculum. Curriculum is being defined as – "In formal education, a curriculum (plural curricula) is the set of courses, and their content, offered at a school or university", while syllabus is defined as "A syllabus is an outline and summary of topics to be covered in a course". They have identified that in CBSE, the advantage is that the curriculum is same all over the country and the continuity of education is not a problem if someone needs to change a school. They have also inferred that ICSE syllabus is tougher than that of the CBSE and state based school boards. Their research work showed that the school boards are giving high importance to evaluation and examination system which includes some additional forms of evaluation such as (a) project work, (b) reading and writing skills, (c) participation in co-curricular activities, (d) attitude and behaviours, etc. However major emphasis was given on written examination by schools. IBO puts more emphasis on project based and practical work compared to the Indian school boards as it follows a global curriculum all over the world. IBO assessment focuses on what skills the students have learnt or what level of understanding can the students demonstrate. British Council in India, in their report [13] on the Indian school education system provides an overall picture into this large and evolving school education system of India. They remarked that "the present education system in India is guided by different objectives and goals but is based around the policies of yester years." They claim that two important policies of the Government of India—the Sarva Shiksha Abhiyan (SSA) in 2001 and the Right of Children to Free and Compulsory Education (RTE) Act, 2009 have made education priorities rise among common people of India and have been responsible for improvements in educational performance. However, this report does not mention about the challenges faced by the Indian school education system today. Also National Institute of Open Schooling (NIOS) is not covered in these reports.

3. NEED AND SIGNIFICANCE OF THE STUDY

The objective of this research is to find the trends and patterns of cognitive level in secondary school board examination questions. To the best of our knowledge, there is no previous research on a comparative analysis of the cognitive level of questions from three different school boards (with respect to India) and in multiple subjects. This study tried to address this gap through a comprehensive research and provide an accessible and open dataset as a solution to the problem and for future similar comparative research.

4. RESEARCH QUESTION

This study investigates the cognitive levels of the questions used in the English medium school leaving examinations for three national school boards in India administered nationwide from 2011 to 2020. It put forwards the following research question: To what extent do Indian school board school leaving examination questions cover the lower and higher-order cognitive levels of Bloom's Taxonomy?

5. DATASET DETAILS AND DESCRIPTION

5.1 Why are these data useful?

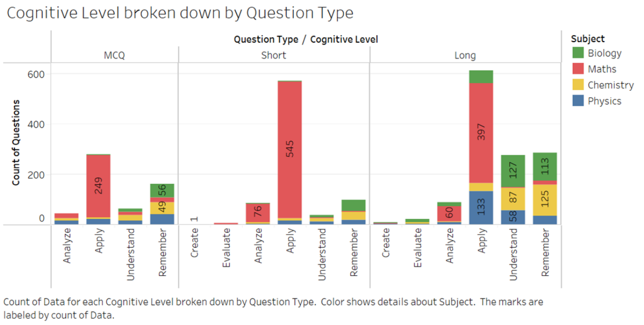
Every year, millions of students appear for the 10th board (secondary exam) in India from various national and regional state education school boards. In 2021, 21,50,608 students appeared in the 10th Board exam in Central Board for Secondary Education Examination (CBSE) in India [1]. However, not much is studied about what Cognitive areas the board exams test, which sections of learning they stress, and which ones they ignore. This research tries to provide a comprehensive comparative analysis of three English medium national school boards – Indian Certificate of Secondary Education (ICSE), the Central Board of Secondary Education (CBSE), and the National Institute of Open Schooling (NIOS). The Indian Certificate of Secondary Education popularly known as ICSE is an examination conducted by the Council for the Indian School Certificate Examination (CISCE), a private board of school education in India for Class 10. The CISCE board is headquartered in New Delhi. The Central Board of Secondary Education (CBSE) is a national-level board of education in India for public and private schools, controlled and managed by the Government of India. There are more than 27,000 schools in India and 240 schools in 28 foreign countries affiliated with the CBSE. The National Institute of Open Schooling (NIOS), formerly National Open School was established by the Ministry of Human Resource Development of the Government of India in 1989 to provide education to all segments of society with the motive to increase literacy and aimed forward for flexible learning. The NIOS is a national board that administers examinations for Secondary and Senior Secondary examinations similar to the CBSE and the ICSE. NIOS enrolls about 350,000 students annually which makes it one of the largest open schooling systems in the world.

5.2 Who can benefit from these data?

This data will help school education stakeholders to get an insight into the cognitive level trend/patterns of assessment questions.This data can be extremely useful and can be further re-used for creating intelligent tools as mentioned in similar research articles like [10] and [9]. The following table, Table 1 shows the details of the dataset.

Table 1: Dataset details

Sl.No.	Board	Years	No. of Ques.
1	ICSE	2011-2020	838
2	CBSE	2011-2020	1274
3	NIOS	2011-2020	959
Total			3071



Cognitive Level broken down by Question Type

Count of Data for each Cognitive Level broken down by Question Type. Color shows details about Subject. The marks are labeled by count of Data.

Figure 1: Types of Questions and Cognitive levels

6. METHODOLOGY

The data was collected from the physical and digital copies of the previous question papers. The text data of digital copies were cleaned and curated in online spreadsheet systems (Google Sheets). Once compiled, all the questions were segregated into individual sheets year and subject-wise[1] . Three annotators having sufficient domain knowledge about Cognitive levels using Bloom's Taxonomy action verbs [11] annotated each question individually to identify the most appropriate cognitive level. The inter-annotator agreement for the appropriate cognitive level was in the substantial agreement range (using Fleiss's kappa) [4]. For data analysis and visualization, Tableau [15] was used.

7. RESULTS

7.1 General Findings

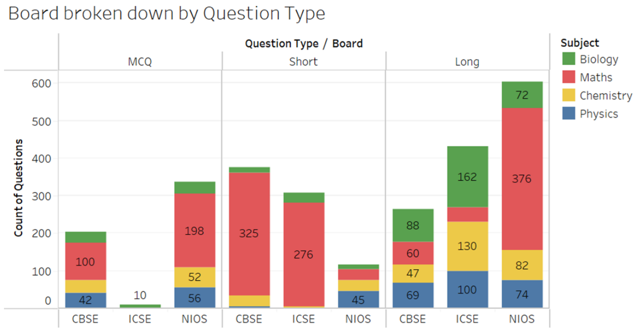
Table 2 and Table 3 shows the top five Bloom's Taxonomy action verbs used for Science and Mathematics across all the three boards. Figure 1 shows the various types of questions, based on their cognitive level and their frequencies. Figure 2 shows various types of questions and their frequencies against each board. Figure 3 shows Cognitive level with respect to individual boards.

7.2 Patterns and Observations

7.2.1 General Question patterns

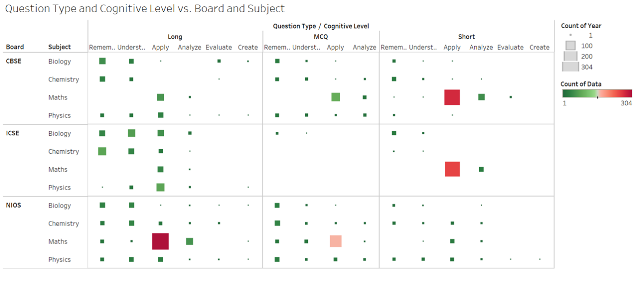
Figure 4 (Mathematics), 5 (Physics), 6 (Chemistry), and 7 (Biology) shows the comparison of the number of questions

[1] CBSE and NIOS takes a combined Physics, Chemistry and Biology exam while ICSE have separate exams for the three subjects.



Count of Data for each Board broken down by Question Type. Color shows details about Subject. The marks are labeled by count of Data.

Figure 2: Types of Questions and School Boards



Count of Data (color) and count of Year (size) broken down by Question Type and Cognitive Level vs. Board and Subject.

Figure 3: A complete overview of School Boards, Question type and Cognitive levels

Table 2: Top 5 Most used Action Verbs (Maths)

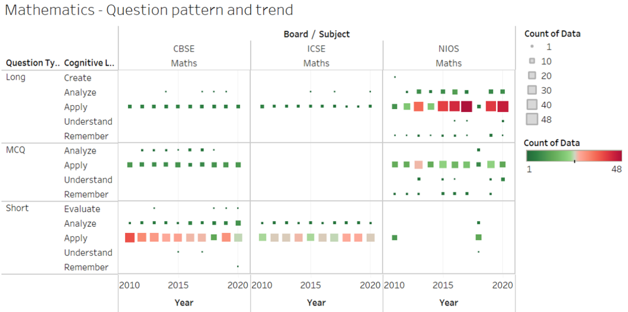
Rank	Verb	fCBSE	fNIOS	fICSE
1	Find	565	220	192
2	Prove	96	36	25
3	Draw	36	9	19
4	Construct	32	18	14
5	Solve	32	11	19

f denotes frequency of verb in respective board.

Table 3: Top 5 Most used Action Verbs (Science)

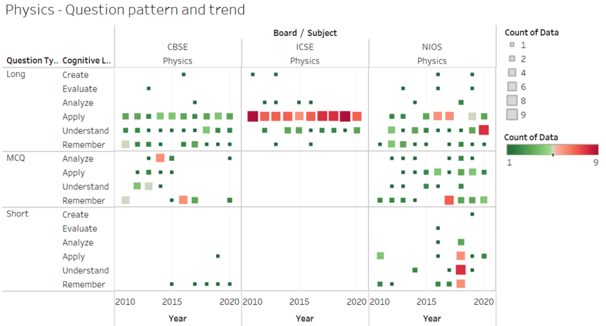
Rank	Verb	fCBSE	fNIOS	fICSE
1	Name	31	31	188
2	Write	68	45	96
3	Explain	44	63	92
4	State	47	32	92
5	Draw	44	30	94

f denotes frequency of verb in respective board.



Year for each Cognitive Level broken down by Board and Subject vs. Question Type. Color shows count of Data. Size shows count of Data. Details are shown for Subject. The view is filtered on Subject, which keeps Maths.

Figure 4: Question Pattern - Mathematics

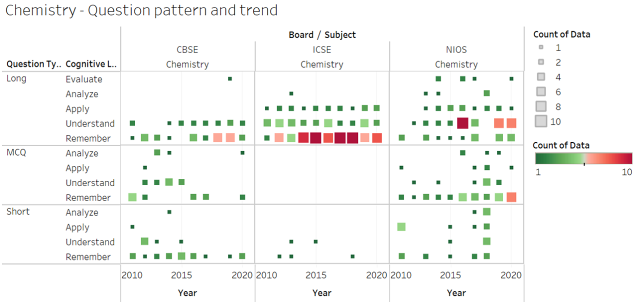


Year for each Cognitive Level broken down by Board and Subject vs. Question Type. Color shows count of Data. Size shows count of Data. Details are shown for Subject. The view is filtered on Subject, which keeps Physics.

Figure 5: Question Pattern - Physics

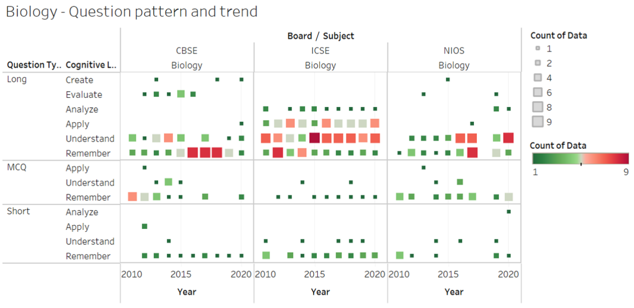
asked on behalf of different Cognitive levels of Bloom's Taxonomy against each Board. Figure 8 shows an overall comparison of Boards, Question type and Cognitive levels. From this analysis, it can be inferred which type of assessment was most preferable for a specified cognitive level. As it is observed, Application-level questions are most frequently asked in all three types of questions (MCQ, Long, Short).

For the MCQ type of questions, there was no "create" and "evaluate" type of questions asked. The reason can be that MCQ questions are known as single marks questions, and asking for creation and evaluation will be much more in



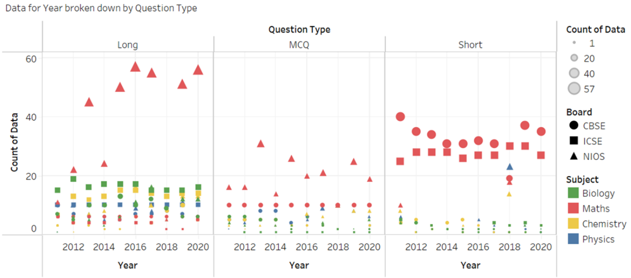
Year for each Cognitive Level broken down by Board and Subject vs. Question Type. Color shows count of Data. Size shows count of Data. Details are shown for Subject. The view is filtered on Subject, which keeps Chemistry.

Figure 6: Question Pattern - Chemistry



Year for each Cognitive Level broken down by Board and Subject vs. Question Type. Color shows count of Data. Size shows count of Data. Details are shown for Subject. The view is filtered on Subject, which keeps Biology.

Figure 7: Question Pattern - Biology



The plot of count of Data for Year broken down by Question Type. Color shows details about Subject. Size shows count of Data. Shape shows details about Board. The view is filtered on count of Data, which ranges from 1 to 75.

Figure 8: Question Pattern - Overall

terms of time and effort required. In long and short type questions, all Cognitive levels of Bloom's Taxonomy questions can be found. It was also observed that lower cognitive levels of Bloom's Taxonomy questions are asked more for the long answer types questions.

7.2.2 *Board vs Subject vs Cognitive level*

The highest number of application-level questions are asked in Mathematics subject in all three boards. Physics also had the highest frequency of Application level questions among all three boards similar to Mathematics. For Biology subject most number of Knowledge level questions were asked for CBSE and NIOS boards. At the same time, ICSE emphasises more on the understanding-type questions, as they had the highest tally among all other cognitive levels. Similarly for Chemistry, Knowledge level questions frequency was highest among all three board subjects. For all three boards for 9 years, there were no create level cognitive-type questions asked in Chemistry.

7.3 Discussions

7.3.1 *Analysis of the three boards for educational and assessment practices*

According to the study findings, lower order cognitive level exam questions outnumber higher order cognitive level exam questions by a wide margin. Such an imbalance in questions based on the six cognitive domains in national examinations may have a negative impact on instructional quality and student learning. CBSE has two categories of papers in almost every year, Delhi and Outside Delhi. These papers have some differences in number of questions in each cognitive level. In ICSE science, all questions have multiple subparts (upto 5 or 6) each from different chapters as well as cognitive levels. These can be very confusing for the student and can cause ambiguities in their analysis. Maths papers throughout the three boards have an abundance of application level questions. Most of these have the somewhat ambiguous Bloom's Taxonomy Action Verb 'Find'. In NIOS papers (Mathematics), the questions for visually impaired students are usually of lower cognitive level than the equivalent for the other students. NIOS Maths asks students to name and define terms in the form of MCQs (approximately 5%) that comes under memorization level and is absent in CBSE. CBSE papers before 2018 only test concepts from the second half of the syllabus, as was the case in Continuous And Comprehensive Evaluation (CCE) pattern.

7.3.2 *The impact of secondary school national board exams on teaching and learning quality*

The questions indicated that students are required to spend more instructional time preparing for the exam or studying past exams that are heavily lower cognitive thinking (LOTS) driven and derived from curriculum books. As a result, students will fail to master complex reasoning skills as required by the curriculum. Students are more likely to face challenges in secondary and tertiary education, as well as in their personal and professional lives, if primary education does not include critical thinking instruction and long-term assessment.

8. LIMITATIONS OF THE STUDY AND FUTURE WORK

The study, however, was limited to the English medium national school board examinations. With 30 or more state boards in regional languages, data collection and annotation would have been a real challenge. However, if it had been implemented, it would have provided a more in-depth understanding of the national education system by reflecting on assessment approaches.

9. CONCLUSION

All the Indian school boards are similar in terms of making students remember the facts, understand the concepts and apply them to solve problems. Thus the questions for evaluation and assessment also focus on these three major aspects. The CBSE syllabus is comparatively "more easy" on students in its approach as it has been designed for a specific year and is divided into various segments. Every segment is given a specific number of periods so that it can be completely and thoroughly taught in one year. It emphasises on understanding of concepts and processes with their application. The ICSE system stresses more in terms of aptitude development and thoroughness by almost equally focusing the syllabus on remembering facts, understanding of concepts and application of the processes learnt. NIOS being an open schooling option, do provide some relaxation in terms of cognitive rigour when compared to the other two national boards. In conclusion we can say, if the syllabus is written considering the Bloom's Taxonomy and knowledge dimension, it will be lot easier to analyze and evaluate that whether the learning objectives have been successfully satisfied on completion of the curriculum.

10. REFERENCES

[1] Cbse 10th results 2021. https://www.ndtv.com/education/cbse-10th-result-2021-9904-pass-result-under-process-for-over-16000-students. [Accessed 14-Jan-2023].

[2] Rigor/Relevance Framework: A Guide to Focusing Resources to Increase Student Performance — leadered.com. https://leadered.com/papers/rigor-relevance-framework-a-guide-to-focusing-resources-to-increase-student-performance/. [Accessed 20-Jan-2023].

[3] L. W. Anderson. Objectives, evaluation, and the improvement of education. *Studies in educational evaluation*, 31(2-3):102–113, 2005.

[4] R. Artstein. Inter-annotator agreement. In *Handbook of linguistic annotation*, pages 297–313. Springer, 2017.

[5] I. R. Assaly and O. M. Smadi. Using bloom's taxonomy to evaluate the cognitive levels of master class textbook's questions. *English Language Teaching*, 8(5):100–110, 2015.

[6] S. Chadwick. Curriculum development in orthodontic specialist registrar training: Can orthodontics achieve constructive alignment? *Journal of orthodontics*, 31(3):267–274, 2004.

[7] S. Das. Cognitive Level Analysis in a Learning Cycle. In *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)*, pages 449–451, Mumbai, July 2018. IEEE.

[8] S. Das and A. Basu. A sample study on the distribution of indian school curricula over bloom's cognitive domain categories. In *EDULEARN16 Proceedings*, 8th International Conference on Education and New Learning Technologies, pages 4811–4817. IATED, 4-6 July, 2016 2016.

[9] S. Das and A. Basu. A sample study on the distribution of indian school curricula over bloom's cognitive domain categories. In *EDULEARN16 Proceedings*, pages 4811–4817. IATED, 2016.

[10] S. Das, S. K. Das Mandal, and A. Basu. Cognitive complexity analysis of learning-related texts: A case study on school textbooks. In *International Conference in Methodologies and intelligent Systems for Techhnology Enhanced Learning*, pages 74–84. Springer, 2020.

[11] S. Das, S. K. Das Mandal, and A. Basu. Classification of action verbs of bloom's taxonomy cognitive domain: An empirical study. *Journal of Education*, 202(4):554–566, 2022.

[12] N. M. Freahat and O. M. Smadi. Lower-order and higher-order reading questions in secondary and university level efl textbooks in jordan. *Theory & Practice in Language Studies*, 4(9), 2014.

[13] B. C. India. Indian school education system-an overview. retrieved march 20, 2017, 2014.

[14] D. R. Krathwohl. A revision of bloom's taxonomy: An overview. *Theory into practice*, 41(4):212–218, 2002.

[15] A. Ohmann and M. Floyd. *Creating Data Stories with Tableau Public*. Packt Publishing Ltd, 2015.

[16] T. V. Ramirez. On pedagogy of personality assessment: Application of bloom's taxonomy of educational objectives. *Journal of personality assessment*, 99(2):146–152, 2017.

[17] W. Sundayana, P. Meekaeo, P. Purnawarman, and D. Sukyadi. Washback of english national exams at ninth-grade level in thailand and indonesia. *Indonesian Journal of applied linguistics*, 8(1):167–176, 2018.

Can ChatGPT Detect Student Talk Moves in Classroom Discourse? A Preliminary Comparison with Bert

Deliang Wang
Faculty of Education
The University of Hong Kong
Hong Kong
wdeliang@connect.hku.hk

Dapeng Shan
Faculty of Engineering
The University of Hong Kong
Hong Kong
dpshan@cs.hku.hk

Yaqian Zheng
Faculty of Education
Beijing Normal University
China
zhengyq@mail.bnu.edu.cn

Kai Guo
Faculty of Education
The University of Hong Kong
Hong Kong
kaiguo@connect.hku.hk

Gaowei Chen
Faculty of Education
The University of Hong Kong
Hong Kong
gwchen@hku.hk

Yu Lu
Advanced Innovation Center
for Future Education, Faculty
of Education
Beijing Normal University
China
luyu@bnu.edu.cn

ABSTRACT

Student utterances in classrooms contain valuable information related to learning. Researchers have employed artificial intelligence techniques, particularly supervised machine learning, to analyze student classroom discourse and provide teachers and students with meaningful feedback. However, supervised models necessitate manual annotation of data, which is both laborious and time-consuming. Recently, OpenAI has released the pre-trained large language model, ChatGPT, which can engage in conversations and provide human-like responses to prompts. Therefore, this study examines the use of ChatGPT in automatically analyzing student utterances and evaluates its capability in addressing the challenge of manual data annotation. Specifically, we compare the performance of ChatGPT with a Bert-based model in identifying student talk moves in mathematics lessons. The preliminary results indicate that while ChatGPT may not perform as strongly as the Bert-based model, it demonstrates potential in detecting specific talk moves, such as *relating to another student*. Additionally, ChatGPT offers clear explanations for its predictions, resulting in higher interpretability compared to the Bert-based model, which operates as a black box.

Keywords

Classroom discourse, talk move, ChatGPT, Bert.

1. INTRODUCTION

Student utterances in class contain rich information about their communicative goals or actions [4], ideas [5], knowledge states, and abilities [8], which are correlated to learning. To assist teachers in understanding student utterances and providing adaptive teaching, studies have adopted artificial intelligence (AI) techniques to model student utterances. For example, researchers have used Long Short-Term Memory (LSTM) networks to estimate whether students have mastered example questions based on their utterances [2]. However, most studies rely on supervised models, which have a significant limitation. Supervised models typically require researchers to manually label a large amount of data in advance, which is laborious and time-consuming. In addition, the trained models may not be easily generalized to other educational contexts.

With the advancement of natural language processing (NLP) techniques, pre-trained large language models such as BERT [3] and GPT-3 [1] have emerged and have demonstrated strong performance on various downstream tasks. Recently, ChatGPT, the latest large language model from OpenAI, has also gained popularity quickly across the whole world [1]. Based on GPT-3 [1] and InstructGPT [12], ChatGPT can engage in conversations with users and generate human-like text responses based on their prompts, such as debugging code and writing essays, which shows exceptional ability in understanding language and indicates great potential in various tasks.

Thus, this paper investigates the ability of ChatGPT to automatically analyze student utterances in classroom discourse and explores whether it can address the challenge of manually annotating data. Specifically, this paper compares ChatGPT and a BERT-based model in automatically detecting student talk moves (i.e., specific dialogic acts) in mathematics lessons. The experiment results show that the BERT-based model outperforms ChatGPT, but ChatGPT demonstrates potential in detecting specific talk moves. In addition, ChatGPT provides clear explanations for its predictions on student utterances, while the BERT-based model operates as a black box and lacks interpretability.

D. Wang, D. Shan, Y. Zheng, K. Guo, G. Chen, and Y. Lu. Can chatgpt detect student talk moves in classroom discourse? a preliminary comparison with bert. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 515–519, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115772

[1] https://openai.com/blog/chatgpt/

2. RELATED WORK

2.1 Automated Models on Student Discourse

Recently, many studies have employed AI techniques to analyze student discourse and provide feedback for learners and teachers. This can be further divided into offline and online learning based on their educational contexts. In offline learning, researchers have not only explored the use of AI chatbots to support students' learning in multiple subjects such as English [10] and engineering [23] but also leveraged LSTM to detect breakdowns in students' conversations with the chatbot in classrooms [11]. Additionally, they have investigated building a convolutional neural network (CNN) based model to automatically identify the semantic content of student dialogue (e.g., prior knowledge, uptake, and querying) in math, science, and physics lessons [17]. In online learning, researchers have used decision trees and naive bayes to classify learners' speech acts (e.g., statement and request)[15], and utilized a Bert-based model to predict learners' dialogue acts (e.g., question, answer, and statement) in science lessons[9]. Student dialogue in collaborative learning is often analyzed to facilitate their learning. For example, researchers have leveraged transformers to automatically classify the dialogue into cumulative, disputational, and exploratory talk [21], and built learners' knowledge graphs to estimate their knowledge [24]. Additionally, students' emotions (e.g., positive and negative) and their behaviour (e.g., knowledge building and off-topic activities) has also been modeled by Bert-based models [26].

2.2 ChatGPT

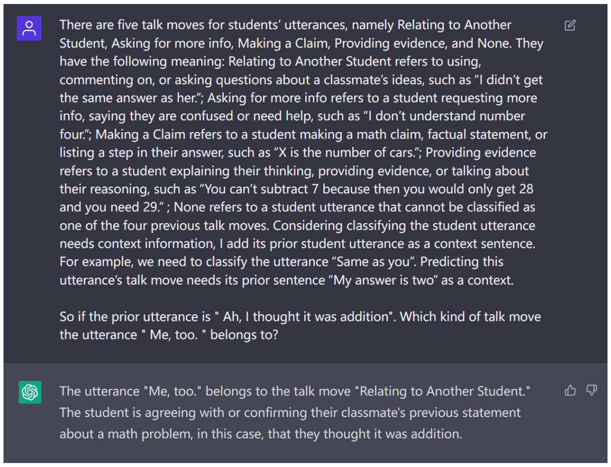
ChatGPT is one of the latest pre-trained large language models developed by OpenAI, which has attracted over 1 million users within 5 days of its release in 2022. Compared to previous language models (i.e., GPT-1 [13], GPT-2 [14], GPT-3 [1]) that may generate harmful and untruthful content, ChatGPT employs the reinforcement learning from human feedback (RLHF) method [18, 12] that changes the training objective from predicting the next token to following human instructions safely, which enables it to generate human-like answers to users' questions. This makes it a powerful tool for various applications, such as composing poetry, commenting on news, and editing language. In the context of education, ChatGPT demonstrates great potential in facilitating learning. For example, users have explored using ChatGPT in language learning (e.g., translating language and providing feedback on writing) [22, 7] and programming learning (e.g., interpreting and debugging code) [20]. As ChatGPT is a relatively new model, there are limited studies examining its use in education. In this paper, we investigate ChatGPT's capability in identifying student talk moves in classroom discourse, to evaluate its potential for providing teachers with effective feedback.

3. METHOD

This section describes how this paper compares the performance of a Bert-based model (i.e., BertForSequenceClassification) and ChatGPT in detecting student talk moves in a dataset.

Table 1: Distribution of student talk moves

Talk Move	Number
Relating to Another Student	353
Asking for more Information	108
Making a Claim	1135
Providing Evidence	664
None	1781



There are five talk moves for students' utterances, namely Relating to Another Student, Asking for more info, Making a Claim, Providing evidence, and None. They have the following meaning: Relating to Another Student refers to using, commenting on, or asking questions about a classmate's ideas, such as "I didn't get the same answer as her."; Asking for more info refers to a student requesting more info, saying they are confused or need help, such as "I don't understand number four."; Making a Claim refers to a student making a math claim, factual statement, or listing a step in their answer, such as "X is the number of cars."; Providing evidence refers to a student explaining their thinking, providing evidence, or talking about their reasoning, such as "You can't subtract 7 because then you would only get 28 and you need 29." ; None refers to a student utterance that cannot be classified as one of the four previous talk moves. Considering classifying the student utterance needs context information, I add its prior student utterance as a context sentence. For example, we need to classify the utterance "Same as you". Predicting this utterance's talk move needs its prior sentence "My answer is two" as a context.

So if the prior utterance is " Ah, I thought it was addition". Which kind of talk move the utterance " Me, too. " belongs to?

The utterance "Me, too." belongs to the talk move "Relating to Another Student." The student is agreeing with or confirming their classmate's previous statement about a math problem, in this case, that they thought it was addition.

Figure 1: An example of the prompt for ChatGPT and its answer.

3.1 Data

In this paper, we selected *TalkMoves* [19], a classroom discourse dataset on K-12 mathematics lessons as our data source. Due to the unavailability of an API interface from OpenAI[2], we were only able to repeatedly utilize ChatGPT for predicting the talk move of a student utterance, which was a time-consuming and challenging task. To address this limitation, we selected a subset from from the *TalkMoves* dataset. Specifically, we chose all primary school lessons in 2021, consisting of 34 transcripts with a total of 4041 student utterances, each of which was annotated with a talk move label. Talk moves refer to specific dialogic acts reflecting the intention of an utterance and speakers' communicative goals [16], and accurately identifying student talk moves is important for teachers to make appropriate response to students. Student talk moves in the *TalkMoves* dataset include *relating to another student*, *asking for more information*, *making a claim*, *providing evidence*, and *None*[19]. The data were not evenly distributed, which can be seen in Table 1. For each type of talk move, we randomly selected 90% of the data as the training set and used the remaining 10% as the testing set. We compared the performance of a Bert-based model and ChatGPT on the testing set.

3.2 Bert-based Model

In this paper, we selected a Bert-based model (i.e., BertForSequenceClassification) as a baseline because its training process (e.g., next sentence prediction) considered the context information [3] and it showed strong performance in text classification tasks [25]. For this specific task of student talk move detection in the TalkMoves dataset, we treated it as a 5-way sequence classification problem. To account for

[2] This work was conducted in December 2022, and the API interface of ChatGPT was made publicly available by OpenAI in March 2023.

Table 2: **Overall performance of the Bert-based model and ChatGPT**

	Bert-based	ChatGPT
accuracy	0.746667	0.582222
precision	0.651488	0.503348
recall	0.561072	0.519613
f1 score	0.599339	0.483108

the importance of dialogue context in identifying talk moves, we set the input of the model as a student utterance concatenated with its preceding utterance. The representation of the input, obtained from the BERT architecture, was fed into a linear layer, and the softmax function was used to predict the talk move. When training the model, we set the learning rate, optimizer, batch size, and number of epochs as 1e-5, AdamW, 32, and 6 respectively.

3.3 ChatGPT

The key to using ChatGPT to detect student talk moves is to provide suitable prompts. We explored several different prompts and selected a suitable one. Specifically, inspired by the idea of few-shot learning from GPT-3 [1], we first provided ChatGPT with the definition and an example of each talk move based on their original description [19]. For example, *Relating to Another Student refers to using, commenting on, or asking questions about a classmate's ideas, such as "I didn't get the same answer as her."* Then, we also clarified the importance of context information, similar to what we did in the Bert-based model. Finally, we asked ChatGPT to predict the talk move of a student utterance. We attempted to provide a batch of student utterances for ChatGPT, but it outputted multiple predictions that did not match the number of inputs in the batch. Thus, we asked ChatGPT to identify the student talk move one utterance by one utterance. An example of the prompt we gave to ChatGPT and its answer can be seen in Figure 1. Considering that ChatGPT may generate inconsistent answers to the same question, this preliminary study adopted the first output as the prediction.

4. RESULT

4.1 Performance

The Bert-based model achieved 0.7523 in F1 score and 0.8164 in accuracy on the testing set. Considering the role of talk moves in improving learning [6], we eliminated student utterances tagged with *None* and only compared the performance of the Bert-based model and ChatGPT in identifying the other four meaningful talk moves, as seen in Table 2. It is evident that the Bert-based model outperforms ChatGPT in accuracy, precision, recall, and F1 score. For instance, the accuracy of the Bert-based model in detecting the four talk moves is around 0.747 while that of ChatGPT is only around 0.58.

Table 3 illustrates the performance of the Bert-based model and ChatGPT in each type of the four talk moves. The Bert-based model performs better in *asking for more information, making a claim, and providing evidence* while ChatGPT achieves stronger performance in *relating to another student.* Additionally, ChatGPT also shows potential in *asking for more information* with 1.0 in the recall metric.

4.2 Interpretability

Despite the superior performance in detecting student talk moves, the Bert-based model is limited in interpretability, as it cannot provide the reason why the prediction is obtained. By contrast, ChatGPT offers clear explanations for each prediction. For example, given a student utterance, "*I did the same thing as Josh did*", the prior student utterance of which is "*I did partial products, and I got it correct*", ChatGPT annotates the utterance with *Relating to Another Student* and gives the following explanations:

> *The student is using and commenting on a classmate's idea (Josh's method, as mentioned in the prior utterance). The student is indicating that they followed the same process as Josh in solving the problem. This talk move involves using, commenting on, or asking questions about a classmate's ideas.*

In this case, ChatGPT points out how the prediction follows the definition of *Relating to Another Student* in the prompt. Another example is that, given a student utterance, "*I'm struggling all the steps in the process.*", the prior utterance of which is "*Okay, we solve it separately and then we all talk about how we each did it?*", ChatGPT labels the utterance with *asking for more information* and gives the following explanations:

> *This talk move is characterized by a student requesting more information or asking for help. In this case, the student is indicating that they are struggling with understanding the steps in the process, which is a request for more information and assistance.*

Similarly, we can see that ChatGPT explains how the prediction is made by analyzing which part of the utterance follows the definition. This feature demonstrates great potential in addressing the interpretability issue of deep learning-based classroom discourse models (e.g., the Bert-based model in this study).

5. DISCUSSION AND CONCLUSION

To automatically analyze classroom discourse without laborious and time-consuming manual annotation of data, the work investigates the capability of the latest large language model, ChatGPT, in identifying student talk moves in mathematics lessons. To achieve this goal, we compare ChatGPT and a Bert-based model in the subset of a classroom discourse dataset. The preliminary results show that although the BERT-based model achieves superior performance, ChatGPT demonstrates the potential in detecting specific talk moves (e.g., *relating to another student*). Specifically, ChatGPT can effectively analyze student utterances that include obvious indicators of talk moves as they align with the definition of the prompt. However, ChatGPT struggles to detect talk moves that are hidden in complex classroom discourse.

In addition, ChatGPT has a significant advantage over the Bert-based model, as it is able to provide detailed and clear explanations for its predictions on student utterances. This feature makes it more interpretable and can increase user

Table 3: Performance of the Bert-based model and ChatGPT in each type of talk move

	Model	Relating to Another Student	Asking for more Information	Making a Claim	Providing Evidence
precision	Bert-based	0.695652	**0.888889**	**0.864078**	**0.808824**
	ChatGPT	**0.727273**	0.458333	0.64486	0.686275
recall	Bert-based	0.457143	0.727273	**0.787611**	**0.833333**
	ChatGPT	0.457143	**1.000000**	0.610619	0.530303
f1 score	Bert-based	0.551724	**0.800000**	**0.824074**	**0.820896**
	ChatGPT	**0.561404**	0.628571	0.627273	0.598291

trust. In contrast, the Bert-based model directly gives predictions without explanations, operating as a black box for users.

As a preliminary study, this exploratory work has several limitations. Firstly, because when the study was conducted, OpenAI did not make the API interface public, the sample size was limited to a relatively small scale, which may cause a bias in the findings. Secondly, as ChatGPT is sensitive to the prompts, changing the prompt may result in different answers. Thus, the choice of prompts may also introduce a bias in the findings. Additionally, it is difficult to determine the optimal prompt for generating the most accurate responses. Thirdly, even if ChatGPT is given the same prompt, it may still generate different answers at different times, which may lead to inconsistency in the results. Fourthly, the study only examines the use of ChatGPT in identifying student talk moves while classroom discourse also carries other valuable information, not limited to talk moves. Besides, teachers' dialogic approach in class can significantly affect teaching and learning. Thus, promising research directions for ChatGPT in classroom discourse include evaluating its ability to identify multiple meaningful characteristics of dialogues between teachers and students in a more extensive dataset with well-crafted prompts and addressing its consistency issue in gnerating answers.

6. ACKNOWLEDGMENTS

This work was supported by Hong Kong Research Grants Council, University Grants Committee (Grant No.: 17605221), and by the Innovation and Technology Commission of the Government of the HKSAR (Grant No.: ITB/FBL/7026/20/P).

7. REFERENCES

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, 2020.

[2] J. Chen, Z. Liu, and W. Luo. Wide & deep learning for judging student performance in online one-on-one math classes. In *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners' and Doctoral Consortium - 23rd International Conference, AIED 2022, Durham, UK, July 27-31, 2022, Proceedings, Part II*, volume 13356 of *Lecture Notes in Computer Science*, pages 213–217. Springer, 2022.

[3] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, pages 4171–4186. Association for Computational Linguistics, 2019.

[4] A. Ezen-Can and K. E. Boyer. Unsupervised classification of student dialogue acts with query-likelihood clustering. In *Proceedings of the 6th International Conference on Educational Data Mining, 2013*, pages 20–27. International Educational Data Mining Society, 2013.

[5] A. Ezen-Can, J. F. Grafsgaard, J. C. Lester, and K. E. Boyer. Classifying student dialogue acts with multimodal learning analytics. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge, LAK '15*, pages 280–289. ACM, 2015.

[6] J. Jacobs, K. Scornavacco, C. Harty, A. Suresh, V. Lai, and T. Sumner. Promoting rich discussions in mathematics classrooms: Using personalized, automated feedback to support reflection and instructional change. *Teaching and Teacher Education*, 112:103631, 2022.

[7] L. Kohnke, B. L. Moorhouse, and D. Zou. Chatgpt for language teaching and learning. *RELC Journal*, page 00336882231162868, 2023.

[8] S. P. Leeman-Munk, E. N. Wiebe, and J. C. Lester. Assessing elementary students' science competency with text analytics. In *Learning Analytics and Knowledge Conference 2014*, pages 143–147. ACM, 2014.

[9] J. Lin, S. Singh, L. Sha, W. Tan, D. Lang, D. Gašević, and G. Chen. Is it a good move? mining effective tutoring strategies from human–human tutorial dialogues. *Future Generation Computer Systems*, 127:194–207, 2022.

[10] K. Mageira, D. Pittou, A. Papasalouros, K. Kotis, P. Zangogianni, and A. Daradoumis. Educational ai chatbots for content and language integrated learning. *Applied Sciences*, 12(7):3239, 2022.

[11] W. Min, K. Park, J. B. Wiggins, B. W. Mott, E. N. Wiebe, K. E. Boyer, and J. C. Lester. Predicting dialogue breakdown in conversational pedagogical agents with multimodal lstms. In *Artificial Intelligence in Education - 20th International Conference, AIED*

2019, Chicago, IL, USA, June 25-29, 2019, Proceedings, Part II, volume 11626 of *Lecture Notes in Computer Science*, pages 195–200. Springer, 2019.

[12] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[13] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[14] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[15] B. Samei, H. Li, F. Keshtkar, V. Rus, and A. C. Graesser. Context-based speech act classification in intelligent tutoring systems. In *International conference on intelligent tutoring systems*, pages 236–241. Springer, 2014.

[16] J. R. Searle. *Speech acts: An essay in the philosophy of language*, volume 626. Cambridge university press, 1969.

[17] Y. Song, S. Lei, T. Hao, Z. Lan, and Y. Ding. Automatic classification of semantic content of classroom dialogue. *Journal of Educational Computing Research*, 59(3):496–521, 2021.

[18] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

[19] A. Suresh, J. Jacobs, C. Harty, M. Perkoff, J. H. Martin, and T. Sumner. The talkmoves dataset: K-12 mathematics lesson transcripts annotated for teacher and student discursive moves. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022*, pages 4654–4662. European Language Resources Association, 2022.

[20] H. Tian, W. Lu, T. O. Li, X. Tang, S.-C. Cheung, J. Klein, and T. F. Bissyandé. Is chatgpt the ultimate programming assistant–how far is it? *arXiv preprint arXiv:2304.11938*, 2023.

[21] S. Ubani and R. Nielsen. Classifying different types of talk during collaboration. In *International Conference on Artificial Intelligence in Education*, pages 227–230. Springer, 2022.

[22] D. Yan. Impact of chatgpt on learners in a l2 writing practicum: An exploratory investigation. *Education and Information Technologies*, pages 1–25, 2023.

[23] C.-C. Yuan, C.-H. Li, and C.-C. Peng. Development of mobile interactive courses based on an artificial intelligence chatbot on the communication software line. *Interactive Learning Environments*, pages 1–15, 2021.

[24] Y. Zhen, L. Zheng, and P. Chen. Constructing knowledge graphs for online collaborative programming. *IEEE Access*, 9:117969–117980, 2021.

[25] L. Zheng, J. Niu, and L. Zhong. Effects of a learning analytics-based real-time feedback approach on knowledge elaboration, knowledge convergence, interactive relationships and group performance in cscl. *British Journal of Educational Technology*, 53(1):130–149, 2022.

[26] L. Zheng, L. Zhong, and J. Niu. Effects of personalised feedback approach on knowledge building, emotions, co-regulated behavioural patterns and cognitive load in online collaborative learning. *Assessment & Evaluation in Higher Education*, 47(1):109–125, 2022.

A Multimodal Language Learning System for Chinese Character Using Foundation Model

Jinglei Yu
School of Educational Technology, Beijing Normal University, China
yujinglei@mail.bnu.edu.cn

Zitao Liu
Guangdong Institute of Smart Education, Jinan University, China
liuzitao@jnu.edu.cn

Mi Tian
TAL Education Group, China
tianmi@tal.com

Deliang Wang
Faculty of Education, The University of Hong Kong, Hong Kong
wdeliang@connect.hku.hk

Yu Lu
Advanced Innovation Center for Future Education, Faculty of Education, Beijing Normal University, China
luyu@bnu.edu.cn

ABSTRACT

Learning Chinese character with multiple definitions is challenging for beginners, while images could help learners get quick understanding and strengthen the memory. To solve the problem, we design a multimodal language learning system for Chinese character featured with AI-generated image definitions. The images with desired semantic meanings are generated by text-to-image foundation model ERNIE-ViLG 2.0. To improve learners' understandings of Chinese character definitions, the system could serve as a knowledge building environment. Learners are expected to contribute ideas collaboratively by voting for the appropriate AI-generated image definitions and choosing to add new qualified ones. The system has been implemented on a mobile application, and future works about estimating and optimizing the built system are discussed.

Keywords

Text-to-image generation, Language learning, Knowledge building

1. INTRODUCTION

Through three thousand years of evolvement, each Chinese character tends to have multiple definitions with original and derived meanings, which is challenging for non-native speakers or even young native speakers to understand and remember. The language and linguistics study found that images could be used as non-verbal mediators, which helps learners build efficient connections between the information and the concepts in memory [8]. In addition, psychologist Allan Paivio proposed dual coding theory [9], which indicates the equal importance of verbal and visual information processing for human, and finds out that visual information could contribute to better memory. Particularly, it has been proved that learners could remember definitions of words better when exposed to both visual and verbal information in second-language learning [10].

We thus propose a multimodal language learning system for Chinese character with both text and image definitions. While the images can be retrieved online, it is hard to guarantee the proper images with the desired meanings could be acquired from the massive online resources. To tackle the issue, we utilize text-to-image generation method to provide the desired images directly from text definitions. Text-to-image generation is one type of AI generation methods, and the cutting-edge enabling technology is based on foundation model (or called pre-trained model) [1]. The capability of foundation model covers language, vision, speech and reasoning, etc. Generally, foundation models are pre-trained on large-scale data and could be flexibly adapted to different downstream tasks via transfer learning, so as to achieve excellent performance. Especially, zero-shot transfer is a feasible way to adapt the model to downstream tasks without tuning parameters. With the help of prompt engineering [7], the foundation model could be fixed and the prompts are used to trigger the model. Conditioned on well-designed text prompt, the text-to-image generation foundation models could create desired and original images. In addition to text-to-image generation, foundation models are capable on many other tasks, such as text-to-text generation (e.g., GPT-3 [2]), image-to-text generation (also known as image caption, e.g., BLIP-2 [6]), text-image pairing (e.g., CLIP [11]), text-to-video generation (e.g., CogVideo [5]), etc.

By leveraging on the foundation model, the system could encourage learners to develop ideas towards the generated images. Specifically, for the same text input, the text-to-image generation foundation model could randomly generate various images. The system supports learners to vote for the appropriate images from all the generated ones. The image with the most votes would be shown at the top of the list for the following learners. In addition, learners could choose to

J. Yu, Z. Liu, M. Tian, D. Wang, and Y. Lu. A multimodal language learning system for chinese character using foundation model. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 520–524, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115764

generate new images and decide whether to add them to the image list as candidates. Based on learners' collaboration, the system could serve as a knowledge building environment to help build community knowledge of Chinese character's definitions.

2. SYSTEM DESIGN

2.1 System Framework and Workflow

The system is a multimodal language learning system for Chinese character featured with AI-generated image definitions. As shown in Figure 1, learners could *input* query Chinese character through user interface, and the system would search Xinhua dictionary's online library via requesting API. Xinhua dictionary, also known as modern Chinese character dictionary, is one of the most authoritative reference books in China. The response is basic text information of the character, including pinyin as phonetic symbol, radical partially indicating semantic meaning, structure representing the stroke composition method, and its multiple definitions. Each definition contains the description of the meaning and its sample words.

Through the user interface, the designed system provides several intelligent functionalities for learners. Firstly, learners could *choose* each text definition to show its image definition. The image definition is directly generated from the text definition by means of foundation model ERNIE-ViLG 2.0. To be specific, ERNIE-ViLG 2.0 [3] is a knowledge-enhanced large-scale Chinese text-to-image diffusion model with 24B parameters, developed by Baidu Inc., China. The diffusion model [4] contains forward and reverse diffusion processes. In forward process, the model gradually adds noises to the image data. While in the reverse process, the model is trained to learn how to denoise and reverse the process to generate the desired image. Based on the basic diffusion model, ERNIE-ViLG 2.0 integrates textual and visual knowledge into the training process to help model focus on important elements, such as critical semantics of texts and salient regions of images. In addition, ERNIE-ViLG 2.0 proposes the Mixture-of-Denoising-Experts (MoDE), which contains multiple "experts" adjusting characteristics of different denoising steps in reverse diffusion process. The performance of ERNIE-ViLG 2.0 is state-of-the-art on text-to-image generation task of zero-shot FID-30K from MS-COCO dataset.

Secondly, learners could also choose to *generate more* images via ERNIE-ViLG 2.0 for the chosen text definition. The newly generated image would show up to the user interface and ask learners to *judge* whether it is appropriate enough to add to the image list. All generated images would be saved to the database as backup, while only the learners confirmed ones could be displayed on the user interface.

Thirdly, learners are encouraged to develop ideas towards the image definitions by voting *like* for the most suitable one. The number of likes would be counted and saved as a key feature of the generated image in the cloud database. The image definitions of the chosen text definition would be displayed on the user interface ranked by the number of likes.

2.2 Text-to-Image Generation Performance

We demonstrate the text-to-image generation performance with an example of Chinese character "Yuan" that has three definitions. As mentioned before, each definition is combined by the description of the meaning and its sample words. The translations of the three definitions of character "Yuan" are shown as the followings:

Definition 1. **Description**: A place where fruits, vegetables, flowers and trees are grown. **Sample words**: Garden. Gardener. Gardening. Garden beds.

Definition 2. **Description**: Originally, it refers to the villa and resting place, and now it refers to the public place for people to play around and entertain. **Sample words**: The Old Summer Palace. Park.

Definition 3. **Description**: Originally, it refers to the tombs of emperors, princes, concubines and princesses of the past generations. **Sample words**: Temple Garden (the ancestral temple built in the graveyard of the emperor). Mausoleum (the tomb of the emperor).

Since the text-to-image generation requires well pre-trained foundation model and efficient computing resource for model inference, we take advantage of Baidu ERNIE-ViLG 2.0 API, and build local server to pre-process text prompt and request the API. The construction of the text prompt is important to the text-to-image generation model, which generally requires two main parts, namely painting object and painting style.

For the painting object part, we investigate two categories of text prompts with help of characters' definitions, which are *description only* and *both description and sample words*. Taking the **Definition 2** of character "Yuan" as an example, we request the two categories of text prompt separately. The results are shown in Figure 2, where Figure 2(b) shows more proper results with *both description and sample words* as text prompt. To be specific, the presentation of Figure 2(a) focuses on the non-critical word "villa" from the description, while Figure 2(b) gets well understanding of both "park" from sample words and "public place for people to play around and entertain" from the description. It may because the description tends to be abstract, while the sample words could provide more specific hints.

For the painting style part, in addition to the realistic style utilized in Figure 2, we also explore various artistic styles like surrealism, conceptual art, impressionism, and different production styles like computer graphic style, illustrator style and pixel style, as shown in Figure 3. Considering about the generalization issues for various Chinese character, we set realistic style for all the image generation, but the system designers or even learners could also make their own choices if needed.

Finally, we identify both description and sample words for painting object part and realistic style for painting style part to construct the text prompt. Four exemplary generated images corresponding to **Definition 1** and **Definition 3** of character "Yuan" are shown in Figure 4.

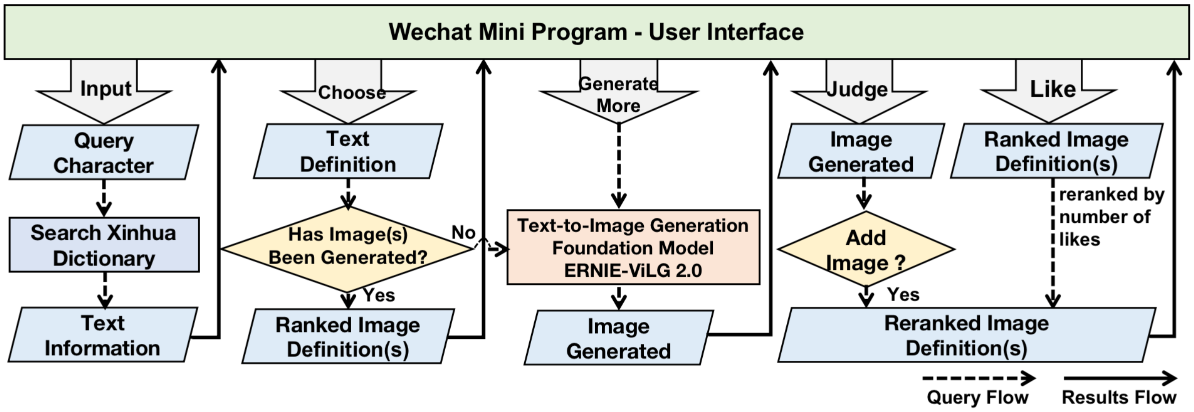
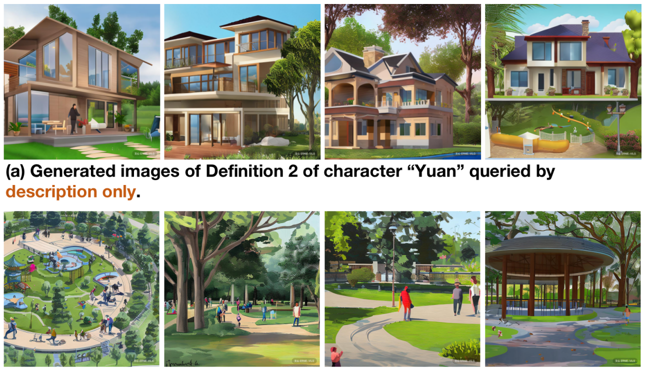

Wechat Mini Program - User Interface
Input
Query Character
Search Xinhua Dictionary
Text Information
Choose
Text Definition
Has Image(s) Been Generated?
No
Yes
Ranked Image Definition(s)
Generate More
Text-to-Image Generation Foundation Model ERNIE-ViLG 2.0
Image Generated
Judge
Image Generated
Add Image ?
Yes
Like
Ranked Image Definition(s)
reranked by number of likes
Reranked Image Definition(s)
Query Flow
Results Flow

Figure 1: System Framework and Workflow



(a) Generated images of Definition 2 of character "Yuan" queried by description only.

(b) Generated images of Definition 2 of character "Yuan" queried by both description and sample words.

Figure 2: Image Generation with Different Painting Objects of Definition 2 of Character "Yuan"

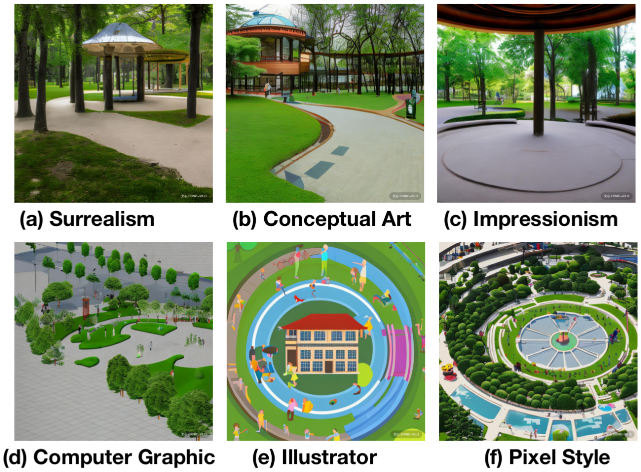


Figure 3: Image Generation with Different Painting Styles of Definition 2 of Character "Yuan"

2.3 Collaborative Learning

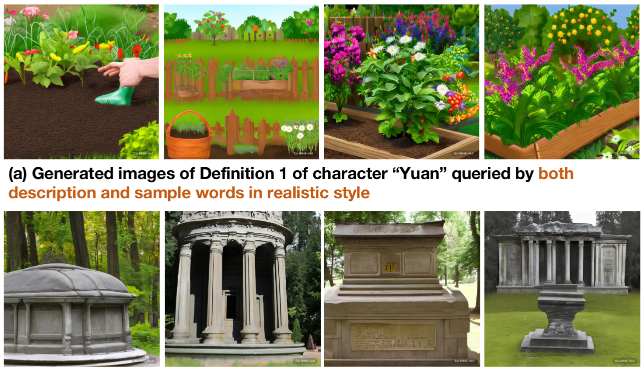
Based on the text-to-image generation results of the foundation model, the system supports learners to collaboratively refine the image definitions. As shown in Figure 2(b) and Figure 4, the image definitions of three text definitions of character "Yuan" are equally important to learners, which demands a high cognitive load to understand them all. To improve the learners' understandings of character definitions, the system encourages learners to vote for the most suitable images based on their understandings, and add new images as candidates when none of the generated images are favored. Ideally, the image with the most votes would be displayed at the top of the list on the user interface and would be considered as the most appropriate image definition to the text definition based on collective knowledge.

The voting and adding image processes require learners to review the text definition carefully and figure out the key semantic meaning of the AI-generated image. Comparing the similarity between the text and image definitions in mind, learners could strengthen the comprehension of the character via verbal and visual dual-channels before making the rational voting decision. When new learners searching the same character, the previous work would support them understanding the text definitions accompanied with most relevant images ranked by others' votes. Meanwhile, new learners could also be inspired to progressively make contributions to the system and work collectively to develop the community knowledge.

3. USER INTERFACE

The user interface of the system is based on WeChat mini program which is a mobile application accessed through WeChat, the most popular social software in China, without extra downloading. Learners could operate it on mobile devices wherever in formal or informal learning environment. As shown in Figure 5, learners could input the query Chinese character in the search box and click on the search button. The system would then return basic information with multiple text definitions of the query character.

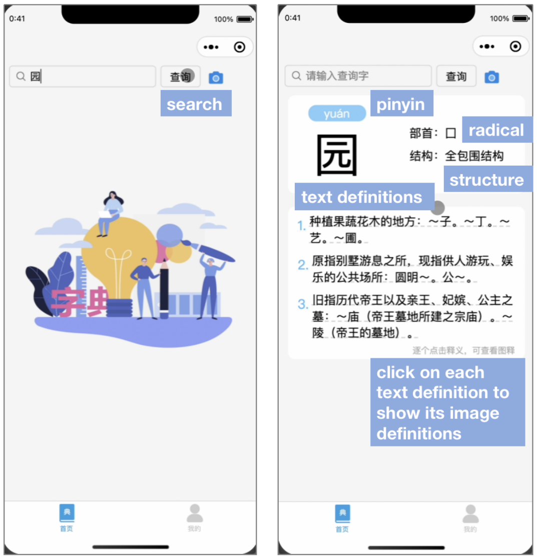
After that, as shown in Figure 6, learners could click on each text definition to show the corresponding image definitions, where the generated images are ranked by the number of likes voted by other learners. It requires learners to browse the generated images from the top-ranked to the bottom,



(a) Generated images of Definition 1 of character “Yuan” queried by both description and sample words in realistic style

(b) Generated images of Definition 3 of character “Yuan” queried by both description and sample words in realistic style

Figure 4: Generated Images of Definition 1 and 3 of Character “Yuan”



部首：口

结构：全包围结构

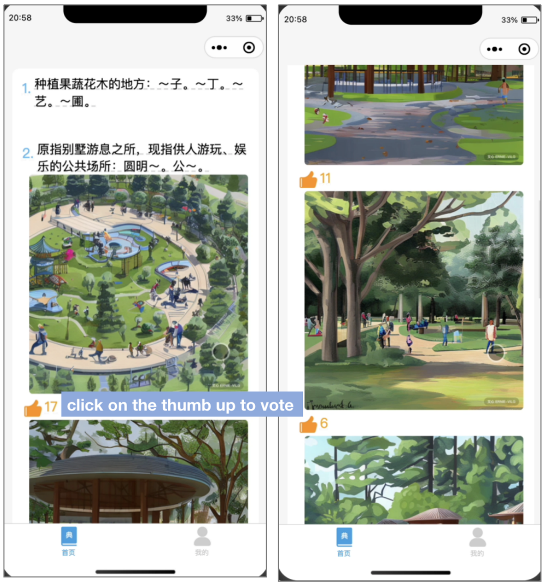
1. 种植果蔬花木的地方：～子。～丁。～艺。～圃。
2. 原指别墅游息之所，现指供人游玩、娱乐的公共场所：圆明～。公～。
3. 旧指历代帝王以及亲王、妃嫔、公主之墓：～庙（帝王墓地所建之宗庙）。～陵（帝王的墓地）。

逐个点击释义，可查看图释

(a) Input Query Character

(b) Search Results

Figure 5: User Interface of Character Querying

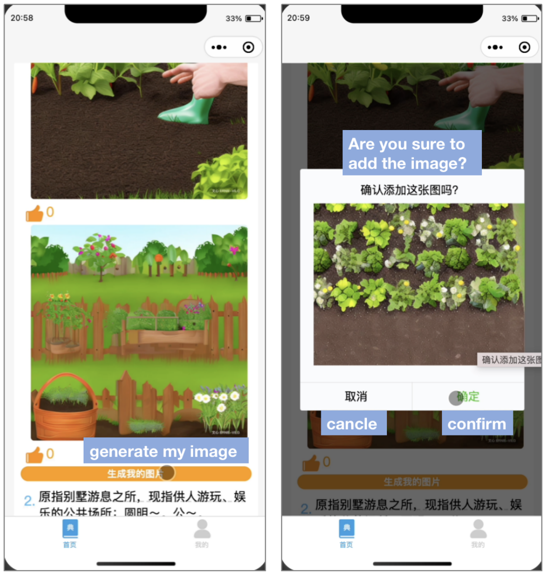


1. 种植果蔬花木的地方：～子。～丁。～艺。～圃。
2. 原指别墅游息之所，现指供人游玩、娱乐的公共场所：圆明～。公～。

(a) Browse Generated Images

(b) Vote for Appropriate Images

Figure 6: User Interface of Images Browsing and Voting



确认添加这张图吗?

2. 原指别墅游息之所，现指供人游玩、娱乐的公共场所：圆明～。公～。

(a) Generate New Image

(b) Confirm to Add the Image

Figure 7: User Interface of Images Generation and Adding

and make their own decisions to vote for the appropriate images by clicking on the thumb up button.

When none of the generated images suitable for the text definition, learners could choose to generate new image by clicking on the “generate my image” button at the bottom of the list, as shown in Figure 7. It takes around 10-20 seconds to generate an image with resolution of 1024 × 1024 pixels. Before adding to the list, a popup would ask for learner’s confirmation, which expects the learner to review the text definition and make a deliberate decision for the image definition.

4. CONCLUSION AND FUTURE WORK

We propose a multimodal language learning system for Chinese character with the help of text-to-image generation foundation model ERNIE-ViLG2.0. Based on the text-to-image generation results, learners could help to improve others’ understandings of Chinese character definitions by voting and adding images to re-rank the images’ display order. Consequently, learners could benefit from the top-ranked images for each character’s text definition and improve the cognition through both verbal and visual channel.

In the future work, to estimate the effectiveness of the system, we plan to design and conduct experiments by inviting entry-level Chinese learners to evaluate their learning achievement and attitudes towards the generated images and the voting system. Especially, it is also worth to investigate the effectiveness of various style images and how they provide improvement in the learning process. Besides, considering about the quality of generated images, the trust of the voting system requires further supervisions to correct typical mistakes from beginning learners and avoid unfriendly attacks.

Additionally, more flexible functions could be added to the built system. For example, in addition to "like" button, "dislike" could also be an option to express learner's opinions on the image. Further, to deepen learners' understandings, it also welcomes learners to make text comments on the image and leave nicknames and avatars to improve community awareness. Besides, since foundation models are pre-trained on large-scale data by black-box method, it is also necessary to require interventions to avoid risks of algorithm biases and intellectual property issues.

Furthermore, the multimodal language learning system could also be transferred to other languages learning with the similar mechanisms of text-to-image generation and learners' collaboration. Additionally, foundation models for AI generation are also powerful on text-to-text generation, image-to-text generation, image modification, etc. It would be interesting to investigate more possibilities of interaction and integration with AI-generated content and learner-generated content.

5. ACKNOWLEDGMENTS

This work was supported in part by National Key R&D Program of China, under Grant No. 2020AAA0104500; in part by National Natural Science Foundation of China (Grant No. 62077006).

6. REFERENCES

[1] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] Z. Feng, Z. Zhang, X. Yu, Y. Fang, L. Li, X. Chen, Y. Lu, J. Liu, W. Yin, S. Feng, et al. Ernie-vilg 2.0: Improving text-to-image diffusion model with knowledge-enhanced mixture-of-denoising-experts. *arXiv preprint arXiv:2210.15257*, 2022.

[4] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[5] W. Hong, M. Ding, W. Zheng, X. Liu, and J. Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.

[6] J. Li, D. Li, S. Savarese, and S. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[7] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[8] R. Oxford and D. Crookall. Vocabulary learning: A critical analysis of techniques. *TESL Canada journal*, pages 09–30, 1990.

[9] A. Paivio. *Mental representations: A dual coding approach.* Oxford University Press, 1990.

[10] J. L. Plass, D. M. Chun, R. E. Mayer, and D. Leutner. Supporting visual and verbal learning preferences in a second-language multimedia learning environment. *Journal of educational psychology*, 90(1):25, 1998.

[11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

Help Me Read! Expanding Students' Reading with Wikipedia Articles[*]

Arun-Balajiee Lekshmi-Narayanan, Khushboo Thaker, Peter Brusilovsky, Jordan Barria-Pineda
School of Computing and Information
135 N Bellefield Avenue
Pittbsurgh, PA, USA
{arl122,kmt81,peterb,jab464}@pitt.edu

ABSTRACT

In this demo paper, we present an implementation of an intelligent digital textbook integrated with external readings for students, such as Wikipedia articles. Our system applies the ideas of concept extraction from a digital textbook on topics in cognitive psychology and computer science for a graduate class in a large US-based university to generate search terms that can link with Wikipedia articles. Finally, we integrate these articles into the textbook reading interface, enabling students to quickly refer to Wikipedia articles in connection with the reading material of the course to understand a concept or topic that they struggle with or are interested in exploring further. With this demo, we present a system that can be utilized for data collection in a real-world classroom setup.

Keywords

Intelligent Textbooks, Digital Reading Systems, Wikipedia, Concept Extraction, Data Collection

1. INTRODUCTION

The rapid development of science and technology created a problem for college instructors who want to ensure that students receive up-to-date knowledge of the subject. While in the past, textbooks served as a predominant source of class readings, they frequently lagged behind the state-of-the-art. At present, many courses, especially at the graduate level, use a collection of recent research papers rather than textbooks as course readings. Unlike textbooks, which introduce domain knowledge gradually, taking care to explain critical concepts, research papers are written for audiences who are already familiar with core domain knowledge. Hence, research papers are challenging to read for unprepared students. Several authors have suggested that recommending relevant Wikipedia articles to explain complicated concepts could facilitate reading [1, 4]. Moreover, as an added benefit, the recommendations could make reading more personalized by encouraging students to explore readings related to their interests. However, implementing Wikipedia recommendations is not straightforward, since only some of the "concepts" mentioned in a research paper are useful recommendations in the context of a specific course. In this demo, we present a course reading system for research papers that uses advances in text mining to recommend the most relevant Wikipedia pages for every page of assigned readings. The system was tested in a full-term graduate course, where we also collected student feedback on the relevance and difficulty of recommended Wikipedia articles.

2. A READING SYSTEM WITH WIKIPEDIA RECOMMENDATIONS

To explore the opportunity to extend online reading with Wikipedia articles, we modified an online digital textbook reading platform, ReadingMirror [2], customizing it to research paper readings. The modified system inherited several useful features from the digital textbook platform, such as a table of contents (now *course reading plan*), annotations, and social comparison (Fig. 1). To extend the reading system with the recommendations of Wikipedia articles, we used text mining to extract entities from each reading page (see Section 3). Page-level extraction was used to provide recommendations on the page where the relevant concept is mentioned. Recommendations are provided using an expandable tab on a page margin. Clicking on this tab reveals a list of links to recommended articles which could be opened next to the article page. For example, if a page of an assigned article mentions "Allen Newell", it is recognized as a useful Wikipedia concept and a link to the Wikipedia article is offered on Allen Newell, along with other recommendations for further exploration and reading (Fig. 2).

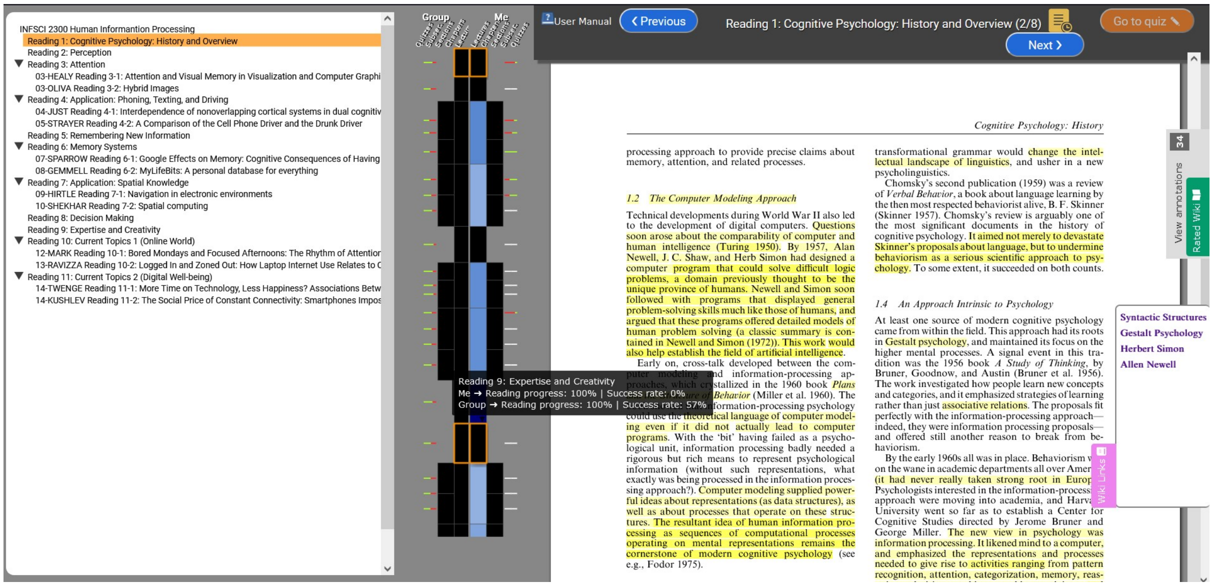
To instrument the classroom study reviewed below, all student work with recommendations (opening, scrolling, and closing the recommendation tab) is logged. In addition, we provide a simple interface for students to rate videos on the relevance and difficulty of recommended Wikipedia articles (bottom left in Fig. 2). To encourage ratings, the list of Wikipedia articles that the student has rated or read appears in a separate tab above the Wikipedia links tab.

3. ENTITY EXTRACTION

Previous work on Wikipedia linking compared the content of the page in the textbook that the student reads with the rel-

A.-B. –Lekshmi-Narayanan, K. Thaker, P. Brusilovsky, and J. Barria-Pineda. Help me read! expanding students' reading with wikipedia articles. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 525–528, Bengaluru, India, July 2023. International Educa-tional Data Mining Society.

https://doi.org/10.5281/zenodo.8115759



INFSCI 2300 Human Informantion Processing
Reading 1: Cognitive Psychology: History and Overview
Reading 2: Perception
▼ Reading 3: Attention
03-HEALY Reading 3-1: Attention and Visual Memory in Visualization and Computer Graphi
03-OLIVA Reading 3-2: Hybrid Images
▼ Reading 4: Application: Phoning, Texting, and Driving
04-JUST Reading 4-1: Interdependence of nonoverlapping cortical systems in dual cognitiv
05-STRAYER Reading 4-2: A Comparison of the Cell Phone Driver and the Drunk Driver
Reading 5: Remembering New Information
▼ Reading 6: Memory Systems
07-SPARROW Reading 6-1: Google Effects on Memory: Cognitive Consequences of Having
08-GEMMELL Reading 6-2: MyLifeBits: A personal database for everything
▼ Reading 7: Application: Spatial Knowledge
09-HIRTLE Reading 7-1: Navigation in electronic environments
10-SHEKHAR Reading 7-2: Spatial computing
Reading 8: Decision Making
Reading 9: Expertise and Creativity
▼ Reading 10: Current Topics 1 (Online World)
12-MARK Reading 10-1: Bored Mondays and Focused Afternoons: The Rhythm of Attentior
13-RAVIZZA Reading 10-2: Logged In and Zoned Out: How Laptop Internet Use Relates to C
▼ Reading 11: Current Topics 2 (Digital Well-being)
14-TWENGE Reading 11-1: More Time on Technology, Less Happiness? Associations Betw
14-KUSHLEV Reading 11-2: The Social Price of Constant Connectivity: Smartphones Impos

Reading 9: Expertise and Creativity
Me ➜ Reading progress: 100% | Success rate: 0%
Group ➜ Reading progress: 100% | Success rate: 57%

Reading 1: Cognitive Psychology: History and Overview (2/8)

Cognitive Psychology: History

processing approach to provide precise claims about memory, attention, and related processes.

1.2 The Computer Modeling Approach

Technical developments during World War II also led to the development of digital computers. Questions soon arose about the comparability of computer and human intelligence (Turing 1950). By 1957, Alan Newell, J. C. Shaw, and Herb Simon had designed a computer program that could solve difficult logic problems, a domain previously thought to be the unique province of humans. Newell and Simon soon followed with programs that displayed general problem-solving skills much like those of humans, and argued that these programs offered detailed models of human problem solving (a classic summary is contained in Newell and Simon (1972)). This work would also help establish the field of artificial intelligence.

Early on, cross-talk developed between the computer modeling and information-processing approaches, which crystallized in the 1960 book *Plans and the Structure of Behavior* (Miller et al. 1960). The [illegible] information-processing psychology could use the theoretical language of computer modeling even if it did not actually lead to computer programs. With the 'bit' having failed as a psychological unit, information processing badly needed a rigorous but rich means to represent psychological information (without such representations, what exactly was being processed in the information processing approach?). Computer modeling supplied powerful ideas about representations (as data structures), as well as about processes that operate on these structures. The resultant idea of human information processing as sequences of computational processes operating on mental representations remains the cornerstone of modern cognitive psychology (see e.g., Fodor 1975).

transformational grammar would change the intellectual landscape of linguistics, and usher in a new psycholinguistics.

Chomsky's second publication (1959) was a review of *Verbal Behavior*, a book about language learning by the then most respected behaviorist alive, B. F. Skinner (Skinner 1957). Chomsky's review is arguably one of the most significant documents in the history of cognitive psychology. It aimed not merely to devastate Skinner's proposals about language, but to undermine behaviorism as a serious scientific approach to psychology. To some extent, it succeeded on both counts.

1.4 An Approach Intrinsic to Psychology

At least one source of modern cognitive psychology came from within the field. This approach had its roots in Gestalt psychology, and maintained its focus on the higher mental processes. A signal event in this tradition was the 1956 book *A Study of Thinking*, by Bruner, Goodnow, and Austin (Bruner et al. 1956). The work investigated how people learn new concepts and categories, and it emphasized strategies of learning rather than just associative relations. The proposals fit perfectly with the information-processing approach—indeed, they were information processing proposals—and offered still another reason to break from behaviorism.

By the early 1960s all was in place. Behaviorism w on the wane in academic departments all over Amer (it had never really taken strong root in Europ Psychologists interested in the information-process approach were moving into academia, and Harva University went so far as to establish a Center for Cognitive Studies directed by Jerome Bruner and George Miller. The new view in psychology was information processing. It likened mind to a computer, and emphasized the representations and processes needed to give rise to activities ranging from pattern recognition, attention, categorization, memory, reas-

Syntactic Structures
Gestalt Psychology
Herbert Simon
Allen Newell

Figure 1: The interface of the reading system, Reading Mirror, with the course reading plan on the left and a page of the assigned reading on the right. A tab on the right of the reading page shows a list of recommended Wikipedia articles related to this page.

evant Wikipedia articles [1, 4]. However, these approaches could be noisy and generate relatively few recommendations. Since one of the goals of our project was to explore the feasibility of generating personalized recommendations that could engage students with different interests, we attempted to generate a somewhat excessive number of recommendations targeting the most relevant concepts mentioned on each page. To achieve this goal, we combined automatic concept extraction with heuristic filtering and embedding-based ranking for each reading page.

The first step in this process is to find Wikipedia concepts and entities mentioned on the target page. For each reading page, we extracted the entities mentioned on the page using the DBpedia Spotlight API [1]. DBpedia Spotlight generates a list of entities in the submitted text along with corresponding Wikipedia pages linked to those entities. This list is usually large and noisy, so it requires post-processing. In the first step of post-processing, we filtered this list based on the semantic types of these entities, removing several irrelevant types of entities such as 'Event', 'Website', 'Film', 'Location', and 'Country'. We also removed entities that did not have a corresponding Wikipedia page in English. After the cleaning, we ranked the remaining entities. Since DBPedia Spotlight does not rank entities according to their relevance to the target page, we used the EMBED Rank [3]. For ranking with EMBED rank, we generated embeddings of the text on the page for which the recommendation is generated and the first paragraph in the ranked Wikipedia page. Top-N Wikipedia pages were recommended to the students.

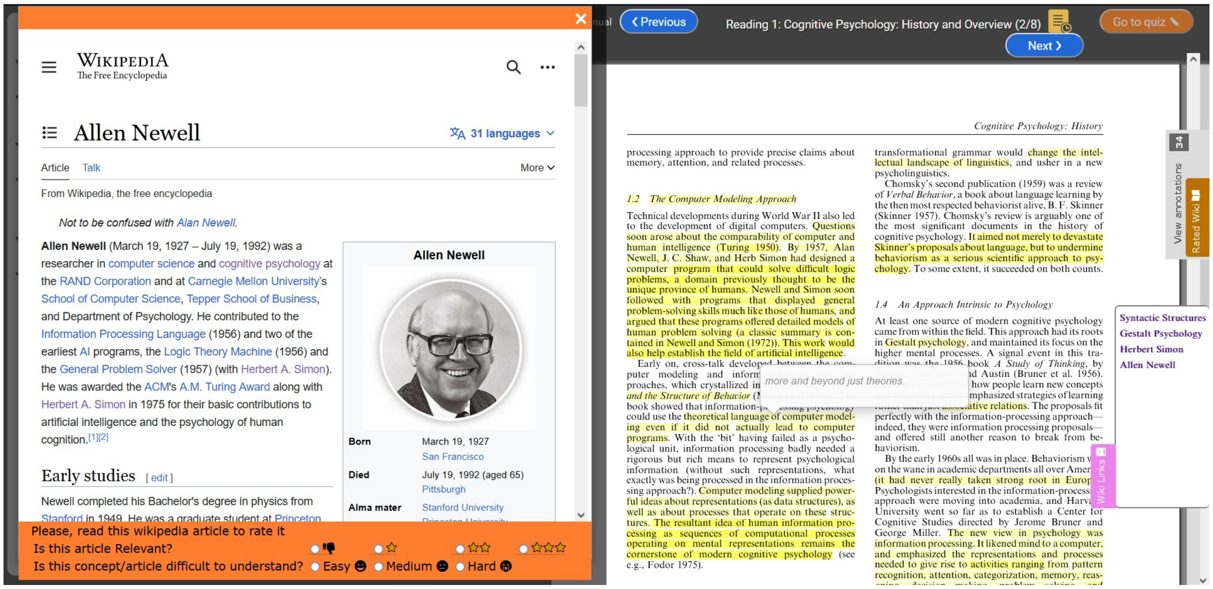
4. A CLASSROOM DEPLOYMENT

To assess the usefulness of our idea and the quality of generated recommendations, we deployed the system as the course reading system in a graduate course on human information processing in a large US-based university. In this lecture-based course, students were requested to read one or two assigned research articles prior to each lecture to prepare for a discussion. In the earlier offerings of this course, the articles were distributed to students in PDF form through a learning management system. In our study, the same articles were provided to students through the course reading system, which allowed us to generate a large number of page–level Wikipedia article recommendations for each assigned research article. The class had 11 lectures with a total of 17 research articles assigned for the required readings. The pages of these articles provided recommendations for 1,238 concepts linked to Wikipedia articles. As part of the learning process, we asked students to read at least 3 Wikipedia articles each week, selecting the most interesting ones for them from the set of recommended articles. In turn, to select these three most interesting articles, students were instructed to examine and rate (by relevance and difficulty) at least 10 recommended articles each week. For this work, students could earn up to one course credit point.

5. PRELIMINARY RESULTS

We collected learning data from 42 students enrolled in the class. In total, 772 out of 1238 recommended concepts linked to Wikipedia articles were explored and rated by students. An average of 12 students ($mean = 12.73$, $std = 8.73$) rated each concept for difficulty and 13 students ($mean = 13.05$, $std = 9.05$) for relevance. The 10 most popular concepts rated for relevance and those rated the most difficult are shown in Table 1. Since the students were guided by their interests, this list likely indicates the concepts in which the students are most interested in the course. Analysis of student rating data indicates that each student rated on average 242 concepts (mean = 241.87, std = 132.12) for difficulty and 242 (mean = 242.97, std = 130.07) for relevance throughout

[1] https://github.com/dbpedia/spotlight-docker



WIKIPEDIA
The Free Encyclopedia

Allen Newell

31 languages

Article Talk More

From Wikipedia, the free encyclopedia

Not to be confused with Alan Newell.

Allen Newell (March 19, 1927 – July 19, 1992) was a researcher in computer science and cognitive psychology at the RAND Corporation and at Carnegie Mellon University's School of Computer Science, Tepper School of Business, and Department of Psychology. He contributed to the Information Processing Language (1956) and two of the earliest AI programs, the Logic Theory Machine (1956) and the General Problem Solver (1957) (with Herbert A. Simon). He was awarded the ACM's A.M. Turing Award along with Herbert A. Simon in 1975 for their basic contributions to artificial intelligence and the psychology of human cognition.[1][2]

Allen Newell

Born	March 19, 1927 San Francisco
Died	July 19, 1992 (aged 65) Pittsburgh
Alma mater	Stanford University

Early studies [edit]

Newell completed his Bachelor's degree in physics from Stanford in 1949. He was a graduate student at Princeton

Please, read this wikipedia article to rate it
Is this article Relevant? ○ 👎 ○ ☆ ○ ☆☆ ○ ☆☆☆
Is this concept/article difficult to understand? ○ Easy ☻ ○ Medium ☻ ○ Hard ☻

Reading 1: Cognitive Psychology: History and Overview (2/8)

Cognitive Psychology: History

processing approach to provide precise claims about memory, attention, and related processes.

1.2 The Computer Modeling Approach

Technical developments during World War II also led to the development of digital computers. Questions soon arose about the comparability of computer and human intelligence (Turing 1950). By 1957, Alan Newell, J. C. Shaw, and Herb Simon had designed a computer program that could solve difficult logic problems, a domain previously thought to be the unique province of humans. Newell and Simon soon followed with programs that displayed general problem-solving skills much like those of humans, and argued that these programs offered detailed models of human problem solving (a classic summary is contained in Newell and Simon (1972)). This work would also help establish the field of artificial intelligence.

Early on, cross-talk developed between the computer modeling and inform proaches, which crystallized in *and the Structure of Behavior* (N book showed that information-processing psychology could use the theoretical language of computer modeling even if it did not actually lead to computer programs. With the 'bit' having failed as a psychological unit, information processing badly needed a rigorous but rich means to represent psychological information (without such representations, what exactly was being processed in the information processing approach?). Computer modeling supplied powerful ideas about representations (as data structures), as well as about processes that operate on these structures. The resultant idea of human information processing as sequences of computational processes operating on mental representations remains the cornerstone of modern cognitive psychology (see e.g., Fodor 1975).

more and beyond just theories.

transformational grammar would change the intellectual landscape of linguistics, and usher in a new psycholinguistics.

Chomsky's second publication (1959) was a review of *Verbal Behavior*, a book about language learning by the then most respected behaviorist alive, B. F. Skinner (Skinner 1957). Chomsky's review is arguably one of the most significant documents in the history of cognitive psychology. It aimed not merely to devastate Skinner's proposals about language, but to undermine behaviorism as a serious scientific approach to psychology. To some extent, it succeeded on both counts.

1.4 An Approach Intrinsic to Psychology

At least one source of modern cognitive psychology came from within the field. This approach had its roots in Gestalt psychology, and maintained its focus on the higher mental processes. A signal event in this tradition was the 1956 book *A Study of Thinking*, by nd Austin (Bruner et al. 1956). how people learn new concepts mphasized strategies of learning ative relations. The proposals fit perfectly with the information-processing approach—indeed, they were information processing proposals—and offered still another reason to break from behaviorism.

By the early 1960s all was in place. Behaviorism w on the wane in academic departments all over Amer (it had never really taken strong root in Europ Psychologists interested in the information-process approach were moving into academia, and Harva University went so far as to establish a Center for Cognitive Studies directed by Jerome Bruner and George Miller. The new view in psychology was information processing. It likened mind to a computer, and emphasized the representations and processes needed to give rise to activities ranging from pattern recognition, attention, categorization, memory, reas-

Syntactic Structures
Gestalt Psychology
Herbert Simon
Allen Newell

Figure 2: Once the student clicks on a link to a recommended Wikipedia article, it opens on the left side of the reading interface. The rating bar at the bottom allows the student to rate the relevance and difficulty of the recommended article.

Table 1: 10 Most Popular Wikipedia articles by number of students rating them as Relevant or Highly Relevant and as Medium or Hard Difficulty

Relevance	Difficulty
Change Blindness	Cognitive Science
Cognitive Science	Memory
Visual Perception	Change Blindness
Cognitive Psychology	Visual Perception
Saccade	Flicker
Experimental Psychology	Saccade
Cognitive Revolution	Cognitive Psychology
Iconic Memory	Distractions
Memory	Metadata
Hybrid Image	Mylifebits

the course duration. Note that it is considerably more than 110 ratings (10 per week) that the students were required to make to get the full score. This data indicates that the students were considerably engaged in examining and rating recommended Wikipedia articles.

The distribution of relevance and difficulty ratings for recommended articles rated is shown in Figure 3. As the data show, the majority of recommended articles were judged easy or medium difficulty by the class, although a noticeable number of articles were considered hard. From the prospect of relevance, the majority of articles were rated as relevant or highly relevant, although a good number were rated somewhat relevant and even not relevant.

To examine the articles rated as relevant or highly relevant, we counted the number of ratings for each of these articles (i.e., the number of students who rated this article as relevant or highly relevant) and plotted this data by ordering articles by the number of ratings (Fig. 4). The data show that while a good number of concepts such as "Cognitive Science" and "Memory" were universally popular, approximately half of the relevant concepts such as "Probabilistic Reasoning" and "Knowledge Visualization" covered in Wikipedia articles were selected for examination by five or fewer students. This confirms our hypothesis that students in the same class have considerably different interests and opens up an opportunity for personalized rather than class-level recommendations.

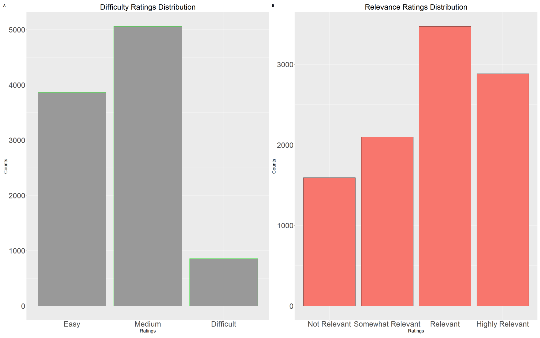


Figure 3: Distribution of Difficulty (left) and Relevance (right) ratings for recommended Wikipedia articles.

As Fig. 3 shows, a considerable number of recommended Wikipedia articles were judged as not relevant. To understand how we can improve the recommendation process, we examined the concepts covered by these Wikipedia articles. The analysis revealed several problems. The dominant source of irrelevant recommendations was the PDF source of research articles. First, hyphenation frequently produces partial words such as "mecha" or "illus", which sometimes have perfectly valid Wikipedia articles unrelated to the content of the course. Second, beyond their true con-

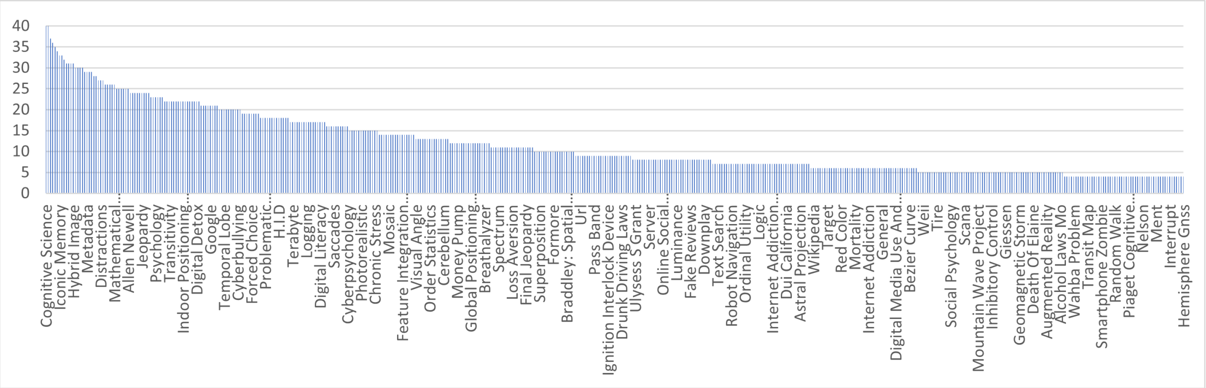


Figure 4: Relevant or highly relevant Wikipedia articles ranked by the number of ratings

tent, all articles have publication data, including named entities for publishers ("Princeton University Press", "SAGE", "IEEE") and places of publication ("Hershey", "Princeton"), which are usually present in Wikipedia. Another problem was the result of our attempt to recognize the names of researchers mentioned in the articles to offer students more information about them. Unfortunately, in a number of cases, these researchers were not prominent enough to appear in Wikipedia, while a different famous person with the same name was listed (i.e., "George Eyser", "Terry Crews"), which resulted in referring to the wrong people. Finally, some perfectly valid concepts such as "priming" (in psychology) had different meanings in different areas and correspond to Wikipedia "disambiguation pages" with links to different meanings. Some students considered these pages irrelevant. The analysis demonstrated that most of the observed problems could be resolved by adding additional heuristics to our filtering process.

6. CONCLUSION

In this demo, we present a system that uses text mining to expand student reading options in graduate classes by recommending relevant Wikipedia articles for research papers assigned for mandatory reading. This approach enriches student course knowledge and allows students to personalize their readings by focusing on the most interesting concepts covered in the recommended articles. The system was used as a primary reading tool in a semester-long graduate course, enabling us to gain several interesting insights into student work with recommendations. In particular, we observed that about half of the articles rated as relevant or highly relevant were examined and rated by 5 or fewer students. It confirms that different students might be interested in different aspects of the course and opens opportunities for personalized recommendations. The current demo used a relatively simple text mining approach to extract interesting concepts mentioned in the text of the mandatory readings, yet the majority of recommended Wikipedia articles (and their concepts) were judged as relevant or highly relevant. The analysis of concepts judged as not relevant revealed several heuristics that could be used to improve our text-mining approach.

7. REFERENCES

[1] R. Agrawal, S. Gollapudi, K. Kenthapadi, N. Srivastava, and R. Velu. Enriching textbooks through data mining. In *Proceedings of the First ACM Symposium on Computing for Development*, pages 1–9, 2010.

[2] J. Barria-Pineda, P. Brusilovsky, and D. He. Reading mirror: Social navigation and social comparison for electronic textbooks. In *First Workshop on Intelligent Textbooks at 20th International Conference on Artificial Intelligence in Education (AIED 2019)*, volume 2225, pages 30–37. CEUR, 2019.

[3] K. Bennani-Smires, C. Musat, A. Hossmann, M. Baeriswyl, and M. Jaggi. Simple unsupervised keyphrase extraction using sentence embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 221–229, Brussels, Belgium, Oct. 2018. Association for Computational Linguistics.

[4] X. Liu and H. Jia. Answering academic questions for education by recommending cyberlearning resources. *Journal of the American Society for Information Science and Technology*, 64(8):1707–1722, 2013.

Characterizing Learning Progress of Problem-Solvers Using Puzzle-Solving Log Data

Haoyu Liu
Stanford University
haoyuliu@stanford.edu

Fan-Yun Sun
Stanford University
fanyun@stanford.edu

Frieda Rong
Stanford University
rongf@stanford.edu

Kumiko Nakajima *
Independent Researcher

Nicholas Haber
Stanford University
nhaber@stanford.edu

Shima Salehi
Stanford University
salehi@stanford.edu

ABSTRACT

The goal of this paper is to gain insight into the problem-solving practices and learning progressions by analyzing the log data of how middle school and college players navigate various levels of Baba Is You, a puzzle-based game. In this paper, we first examine features that can capture the problem-solving practices of human players in early levels. We then examine how these features can predict players' learning progressions and their performance in future levels. Based on the results of the current quantitative analyses and grounded in our previous in-depth qualitative studies, we propose a novel metric to measure the problem-solving capability of students using log data. In addition, we train artificial intelligence (AI) agents, particularly those utilizing Reinforcement Learning (RL), to solve Baba Is You levels, contrast human and AI learning progressions, and discuss ways to bridge the gap between them.

Keywords

Baba Is you, Human Learning, Reinforcement learning, Problem-solving, Learning progression

1. INTRODUCTION

1.1 Problem-solving & Log Data

There is ubiquitous agreement that problem-solving is an important goal of STEM education [8, 4, 3]. However, there is little agreement as to what features compose effective problem-solving or how to teach and measure these features [14]. Advancements in AI and human behavior analysis introduce the possibility of identifying these features, capturing problem-solving performance in rich detail, and consequently providing problem-solvers with just-in-time feedback and scaffolding [16]. Several works have tried to accomplish this using *log data* generated from interaction with a digital environment. For example, Wang et al. [17] have examined how to engineer features from log data to capture the efficacy of problem-solvers' data collection when solving electric circuit problems. Bumbacher et al.[2] and Perez et al. [13] have used log data to determine how deliberately a person engages in problem-solving related to physics. Here we continue this line of work by using log data of problem-solvers interacting with the puzzle-based game Baba Is You. We also compare the problem-solving processes of human problem-solvers with a standard reinforcement learning agent and discuss the potential underlying causes of these differences.

1.2 Reinforcement Learning & Human Comparisons

It has long been noted human learning behaviors in game environments differ significantly from those of standard Reinforcement Learning (RL) algorithms, with much attention paid to the sample inefficiency of the latter [11]. Tsividis et al. [15] study human learning behaviors in the Arcade Learning Environment (commonly referred to as Atari [1]), and hypothesize a range of mechanisms for their differences with RL algorithms. Human and reinforcement learning behavior and attention [9] as well as neural activity [5] have been also compared within the Arcade Learning Environment. Works have investigated the inclusion of object representations [6] and linguistic grounding [10] so as to close the gap between human and RL behaviors. Dubey et al. [7] compare human and RL algorithm behavior in environments specifically designed to limit the usefulness of human visual priors. Our work, while preliminary, eventually seeks to characterize the sorts of representations and motivations RL systems need in order to engage in human-like problem-solving behaviors in challenging problem-solving environments.

2. METHODS

2.1 Baba Is You

Baba Is You is a puzzle game where players can change the rules by which they play. In Baba Is You, players move Baba, a small sheep-like creature, by pressing keys or buttons/joysticks on a controller to make Baba move up, down, left, or right; players can also reverse their actions in a level or restart the level completely. At every level, the rules themselves are present as text blocks that players can inter-

[*] kumiko.nakajima5221@gmail.com

H. Liu, F.-Y. Sun, F. Rong, K. Nakajima, N. Haber, and S. Salehi. Characterizing learning progress of problem-solvers using puzzle-solving log data. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 529–533, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115768

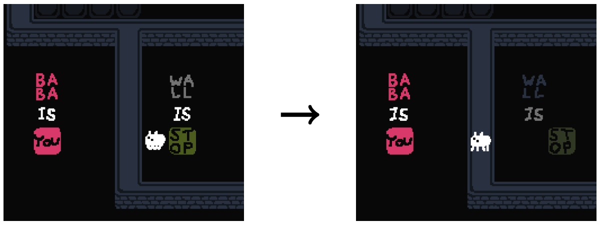


Figure 1: Screenshots from the video game Baba Is You. In this example, Baba can pass through the wall when the "WALL IS STOP" rule is broken (right screenshot).

act with, and by manipulating them, they can change how the level works (see Fig. 1).

The game has various levels, with similar levels grouped into one map. In the beginning, there is a map with 7 tutorial levels that players are required to finish at least 4 of them in order to proceed. Players can then go to the Lake map, which contains 13 normal and 2 extra challenge levels. For all following maps, to unlock the next map, players have to finish at least 8 levels. Map 1 (The Lake) can only be followed by Map 2 (Solitary Island); Map 2 by both Map 3 (Temple Ruins) and Map 4 (Forest of Fall). Finally, after Map 3, players can proceed to all the other maps.

2.2 Participants

Middle school students (n = 54) and college students (n = 113) were recruited via online flyers to participate in the study. We recruited both groups of players to capture a potential range of prior problem-solving expertise. All participants had never played Baba Is You before. Each participant was asked to play the game for three separate sessions. Sessions last up to 150 minutes, and during these sessions, players played as many levels as they wished. They were not required to finish each level they attempted, and they did not have to play a fixed set of levels. Both middle school and college students finished all tutorial levels and some levels from early maps.

2.3 Dataset

2.3.1 *Log Data*

We extracted game log data, which has the timestamps of all player inputs, all game events (e.g., rule-add, rule-remove, no-you) that happened because of player inputs (e.g., left, right, down, up), and the number of the levels completed by each student at any given timestamp.

2.3.2 *Survey Data*

We surveyed participants about their age, grade, and general computer gaming experience, as well as self-reported scores on factors such as approach toward failure and self-efficacy via an online survey after all play sessions ended.

2.3.3 *Aggregated Data*

We created one large aggregated dataset that stores the IDs of the students, their survey answers, the length of their play sessions, the average amount of time they spent on each level, as well as some simple aggregated count features extracted from the log data such as the overall number of restarts, undo.

3. ANALYSIS

3.1 Exploratory Data Analysis

Our goal is to develop a model that predicts student performance on later levels from interpretable variables on earlier levels — such interpretability is crucial for future scaffolding interventions which use this model. Standard feature selection methods from all log data features for this predictive problem may sacrifice such interpretability. Hence, we first explore what features from aggregated data are most predictive of a simplified overall problem-solving progression proxy: the number of levels completed. Then, for predicting future problem-solving performance, we layer in additional level-based features.

3.1.1 *Predicting Overall Problem-solving Progression*

To predict overall problem-solving progression, the aggregated features used are: the number of levels tried, the number of undo inputs, the number of restart inputs, the number of "no you" states (when the player has no controllable representation in the game due to having dismantled "X IS YOU" for all objects X and has no possible moves other than restart or undo), the average session time, the player's game experience level, and their school grade. To examine which features significantly predict the learning progress of students as operationalized by the number of levels completed, we implement k-fold cross-validated linear regression with intercept:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

Here, X denotes any single feature after standardization,(mean 0, sd 1) and Y denotes the number of completed levels. For this k-fold cross-validated linear regression, we use cross-validated R^2 to measure the goodness of fit. For this analysis, we took $k = 10$. We then choose the most important features, as measured by goodness of fit in this analysis, for predicting future problem-solving performance in the subsequent logistic regression analysis as described below.

3.1.2 *Predicting Future Problem-solving Performance*

Because of the small sample size and the distribution of students who tried each level (shown in Fig. 2), we only extract input features from the initial levels for which at least 150 students have attempted (all levels before Lake-9). Then, we build models predicting future performance based on input features from three groups of initial levels: all finished levels before level Lake-9 ('**all-previous**'), the eight hardest levels finished before level Lake-9 ('**8-hardest**'), and from the first and last levels before level Lake-9 ('**first-and-last**'). We use features found to be predictive in the preceding linear regression analysis (Section 3.1.1), and use the performance of students in these features in the selected initial levels to predict students' problem-solving performance in future levels. We use the selected features from overall problem-solving progression prediction as model inputs for this future performance prediction in two different ways: using averages across previous levels, labeled as average values, or including separately all the values of the selected features from the previous levels, labeled as progressive values.

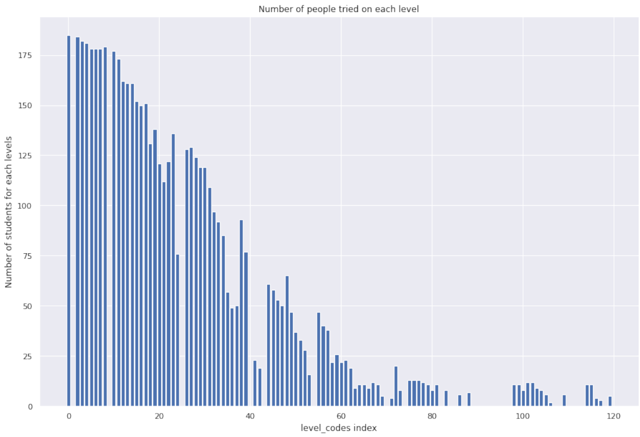


Figure 2: Number of students who attempted each level. Successive levels were attempted less often.

To make future performance prediction tractable, we categorize students into 'high' and 'low' performance and predict these coarse-grained outcomes. To categorize students' performance in a given collection of future levels, we count the number of those levels for which each student finishes within the fastest 50%. We then cut the population along the 50th percentile of this count distribution: students with a higher-than-median count are labeled 1 ('high' performance), and those with lower-than-median count are labeled 0 ('low' performance). Note that these labels depend on the collection of levels for which we predict performance.

We then use logistic regression on these inputs to predict three future problem-solving performance measures separately: performance in the immediate **next level**, performance in **all future levels in the same map**, and performance in **all future levels in a different map**. Logistic regression is used in this analysis as it is a simple and interpretable method that can be effective for binary classification problems, and it does not require any assumptions for the independent variables. Hence, it is a good choice for this preliminary attempt to gain insight into the prediction power and weights of the features. Because of dividing both the performance and the population by median, the random performance of the model (performance by chance) would be 50%. Therefore, we can compare our logistic regression model accuracy relative to this 50% baseline performance. We conduct a shuffle test to make this random prediction rigorous. The shuffle test involves training a model to predict randomly permuted output labels (e.g., high vs. low performance), giving us a "random" model baseline.

Overall, We have then 2 (average or progressive values) * 3 (all-previous, 8-hardest, or first-and-last initial levels) * 3 (to predict performance in the next level, all future levels in the same map, or all future levels in a different map) = 18 fitted logistic regression models in total. For each of these 18 logistic regression models, we run a 10-fold cross-validation on the data-set to estimate generalization of model accuracy. We leave hyperparameters in the default settings for this exploratory analysis.

Table 1: Mean R^2 of linear regression on CV-dataset. no-you count and tried-levels count seems useful for next step's feature extraction

Feature	CV-R^2
no-you count	0.317
tried-levels count	0.470
restart count	0.057
undo count	0.001
Game experience (hour)	-0.187
Avg session time (hour)	-0.159
Age group (Middle school=0, College=1)	-0.136

3.2 Reinforcement Learning

We aim to compare the performance of Reinforcement Learning (RL) agents with human players. We selected three levels (i.e. *baba-is-you*, *out-of-reach*, and *volcano*) with available human play data and trained RL agents on them. The RL agents have a discrete action space of size 4, which includes left, right, up, and down. The state space, or map, is represented by one-hot encodings. For instance, a 6 × 5 environment with 10 distinct tiles would be represented by a floating point tensor in $\mathbb{R}^{10\times6\times5}$.

We implemented a DQN algorithm [12] with an epsilon-greedy strategy to train the RL agents. The discount factor gamma in the DQN algorithm is set to 0.99. The initial exploration rate epsilon is set to 0.9 and decays by a factor of 0.99 after every episode, with a minimum exploration rate of 0.01. The neural network architecture consists of four 2D convolutional layers, followed by batch normalization layers and a linear feed-forward layer. We employed a batch size of 128 and a replay buffer size of 10,000.

For the RL agents, an episode ends when the number of actions taken exceeds 200, when no available action is left, or when the level is solved. In the case of human players, we considered the end of an episode when the player hits the reset button, when no available action is left, or when the level is solved. We devised a reward system to measure the performance of both RL agents and human players. The RL agents receive -100 points for failing to solve the level, +200 points for completing the level, and -0.5 points for each action taken, to incentivize the agent to find more efficient solutions.

To compare human learning progress with the learning progress of RL agents, we scored human play based on the same reward system, even though such a scoring system is not visible to them. As human players typically only solve a level once, we assumed that they can at least repeat their solutions. In our visualizations, we maintained their scores at their top scores after they stopped solving for a particular level.

4. RESULTS

4.1 Important Features

The results of feature selection from aggregated data show that (Table 1) while we initially hypothesized that the no-you event is a major reason that students hit restart, the count of no-you is more important than the count of restart in predicting the number of levels completed. With R^2 =

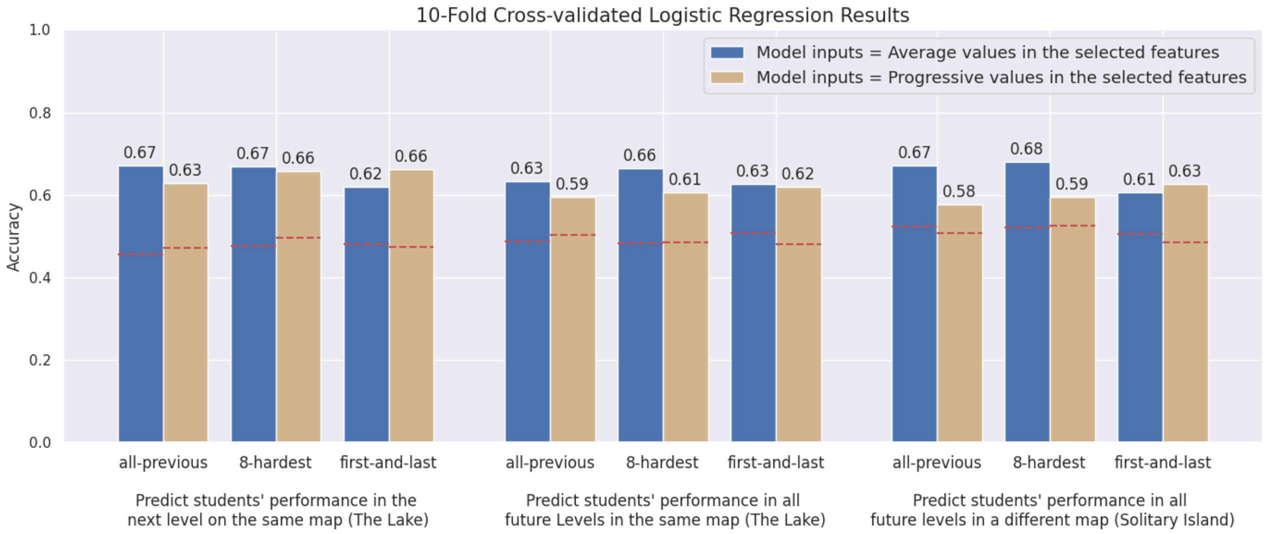


Figure 3: Results (Accuracy) for 10-fold cross-validated logistic regression, with shuffle test baseline (red dash line). The six bars on the right show that using the average performance from all-previous (levels before Lake level 9) completed levels or the eight hardest levels to predict the performance of far levels is accurate. When predicting future-same-map levels' performance, the model that using first-and-last levels from the current map is more accurate

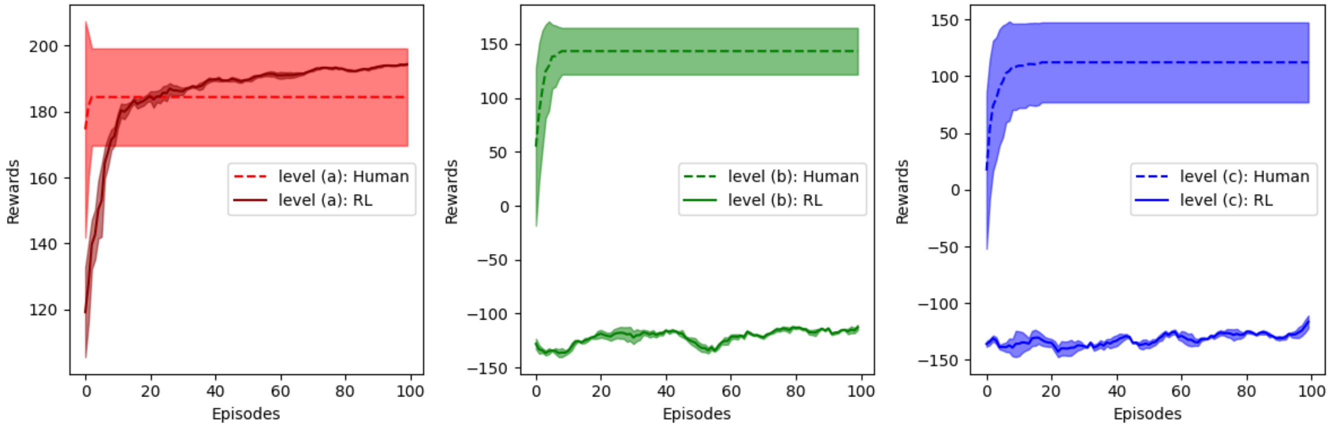


Figure 4: Results for the Reinforcement Learning experiment. The light-colored area shows the standard deviation of all the rewards obtained by all human participants. We find that RL agents can only solve the first level (a), but fail to solve levels (b) and (c), whereas human participants can solve all with much better sample efficiency.

$0.317 < 0.5$, no-you, as a single feature, has a weak predictive power. Thus, we introduce more features directly from the game's log data that improve the model fit significantly, including the count of rules added, count of rules removed, count of a single undo, count of blocks of undos, average time between inputs, maximum time between inputs, and the count of input signals. Overall, the three features that can significantly predict problem-solving progression and the number of initial levels completed are no-you count, tried-level count, and restart count.

4.2 Predicting Future Performance

The results of cross-validated logistic regression to predict future problem-solving performance are shown in Fig. 3. The model that used the features from the first and last previous levels in the current map has reached the highest accuracy when predicting students' future performance in the next single level or all future levels in the same map. Also, there is no significant difference between using average values of the selected features and using progressive values of the selected features. When predicting performance in the levels in a new map, the accuracy of all 3 models (all-previous, 8-hardest, first-and-last) decreases. In addition, it is interesting that for the levels in a new map, using all-previous and 8-hardest models are more accurate than using the first-and-last model while making predictions using the average values of the selected features.

4.3 Reinforcement Learning Comparison

The results of the RL experiment are shown in Fig. 4. Three levels of increasing difficulty (a — Tutorial1, b — Tutorial2, and c — Tutorial3) were chosen for the experiment. It was observed that RL agents were only able to solve level (a), while human players could, on average, solve all levels. Note that a positive reward always indicates that the level has been solved since the only positive reward signal is obtained from solving the level. For level (a), the RL agent was trained to improve its policy, resulting in rewards that increased with more training episodes. Human participants, however, typically only solve each level once. To visually compare the performance of humans with that of the RL agent, we aggregate human performance curves as if they were to continue playing their best score after play had ceased — hence, the human reward curve is flat after episode 10, as most humans solved these levels within 10 episodes. Note that we are plotting these reward curves as a function of episode; as noted in analysis, these definitions differ slightly between human and RL agents. The above plot is our best attempt to compare human learning progress with RL learning progress despite this discrepancy.

5. CONCLUSIONS

Our exploratory regression analysis identifies significant features of human problem-solvers that help them succeed overall in the game as well as help them perform in future levels. We found that features like the number of no-you and undos can predict the problem-solver overall progression. One can hypothesize that the frequency of these features capture the extent that a player explores the game mechanics, and hence impact their overall problem-solving progression in the game. Furthermore, we can predict problem-solving performance in future levels using performance in these features in the hardest previously attempted levels as well as only the first and the last previously attempted levels.

While the RL agent's performance fares similarly to human problem-solvers in an initial level, their performance falls significantly behind in the more challenging levels, and they exhibit significantly different sample efficiency in arriving at solutions. This is entirely expected, as we are training standard RL methods from scratch on these data. One interesting challenge we hope to make progress on is in closing this gap: the sorts of pre-training experience, subsequent representations, agent motivations, and inter-level transfer mechanisms that lead to more human-like problem-solving performance.

6. REFERENCES

[1] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents," arxiv preprint arxiv: 1207.4708. 2012.

[2] E. Bumbacher, S. Salehi, C. Wieman, and P. Blikstein. Tools for science inquiry learning: Tool affordances, experimentation strategies, and conceptual understanding. *Journal of Science Education and Technology*, 27(3):215–235, 2018.

[3] N. R. Council et al. *Learning science through computer games and simulations*. National Academies Press, 2011.

[4] N. R. Council et al. Next generation science standards: For states, by states. 2013.

[5] L. Cross, J. Cockburn, Y. Yue, and J. P. O'Doherty. Using deep reinforcement learning to reveal how the brain encodes abstract state-space representations in high-dimensional environments. *Neuron*, 109(4):724–738, 2021.

[6] C. Diuk, A. Cohen, and M. L. Littman. An object-oriented representation for efficient reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pages 240–247, 2008.

[7] R. Dubey, P. Agrawal, D. Pathak, T. L. Griffiths, and A. A. Efros. Investigating human priors for playing video games. *arXiv preprint arXiv:1802.10217*, 2018.

[8] O. for Economic Co-operation and D. (OECD). *PISA 2012 results: Creative problem solving: Students' skills in tackling real-life problems (Volume V)*. OECD Publishing Pisa, 2014.

[9] S. S. Guo, R. Zhang, B. Liu, Y. Zhu, D. Ballard, M. Hayhoe, and P. Stone. Machine versus human attention in deep reinforcement learning tasks. *Advances in Neural Information Processing Systems*, 34:25370–25385, 2021.

[10] K. Kansky, T. Silver, D. A. Mély, M. Eldawy, M. Lázaro-Gredilla, X. Lou, N. Dorfman, S. Sidor, S. Phoenix, and D. George. Schema networks: Zero-shot transfer with a generative causal model of intuitive physics. In *International conference on machine learning*, pages 1809–1818. PMLR, 2017.

[11] V. Mai, K. Mani, and L. Paull. Sample efficient deep reinforcement learning via uncertainty estimation. *arXiv preprint arXiv:2201.01666*, 2022.

[12] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[13] S. Perez, J. Massey-Allard, D. Butler, J. Ives, D. Bonn, N. Yee, and I. Roll. Identifying productive inquiry in virtual labs using sequence mining. In *International conference on artificial intelligence in education*, pages 287–298. Springer, 2017.

[14] S. Salehi. *Improving problem-solving through reflection*. Stanford University, 2018.

[15] P. A. Tsividis, T. Pouncy, J. L. Xu, J. B. Tenenbaum, and S. J. Gershman. Human learning in atari. In *2017 AAAI spring symposium series*, 2017.

[16] K. D. Wang, J. M. Cock, T. Käser, and E. Bumbacher. A systematic review of empirical studies using log data from open-ended learning environments to measure science and engineering practices. *British Journal of Educational Technology*, 2022.

[17] K. D. Wang, S. Salehi, M. Arseneault, K. Nair, and C. Wieman. Automating the assessment of problem-solving practices using log data and data mining techniques. In *Proceedings of the Eighth ACM Conference on Learning@ Scale*, pages 69–76, 2021.

A Trace-Based Generalized Multimodal SRL Framework for Reading-Writing Tasks

Debarshi Nath
IIT Bombay-Monash Research Academy
debarshi.nath@iitb.ac.in

Dragan Gasevic
Department of Human-Centred Computing
Monash University

Ramkumar Rajendran
Interdisciplinary Programme in Educational Technology
Indian Institute of Technology Bombay

ABSTRACT

Reading-Writing has a ubiquitous presence in almost all kinds of learning. While measurement frameworks for self-regulated learning exist, they are often very contextual and do not guarantee generalizability over more than a specific task. This doctoral project primarily aims to investigate the applicability of a common SRL measurement framework over a range of reading-writing tasks. The research also aims to investigate whether integrating log data, peripheral data like mouse clicks and keystrokes and eyetracking data reveal more information and improve the measurement of SRL.

Keywords

Self-Regulated Learning, Multimodal Analytics, Process Mining, Pattern Mining, Predictive Modelling, Temporal Analytics

1. INTRODUCTION

Writing is an essential part of thinking and learning, whether it be in a school context, in higher education or in a professional setting. Writing tasks is also a critical tool for intellectual and social development [8]. Reading, comprehending, and writing are extremely ubiquitous requirements for all kinds of learning setups. For this reason, developing self-regulation of learners in writing tasks has gained prominence in educational research for a long time [8]. Self-regulation in writing tasks has consequently been explored greatly over the years [8, 12, 1]. But the inception of digital learning environments has opened up new possibilities for understanding learners' mental processes and supporting proper learning strategies through the collection of trace data. Combining trace logs and other forms of multimodal data can reveal more information about learners' latent mental processes and can improve the current state of research [19].

There have been trace-based studies focusing on writing tasks [9, 7, 16, 11, 2]. However, a large number of these studies are very contextual; they are conducted in their ad-hoc learning environments and for their own specific reading-writing task. This statement can actually be made for most SRL-based studies, and rightly so because self-regulated learning is extremely contextual [18]. Most learning environments are so specific that they do not allow generalizations across multiple environments [15]. Researchers do adopt measuring protocols from other studies, but that again raises questions about the validity and reliability of such measurements as such measurements were designed for a very specific learning context.

A learner's adoption of strategies can also depend on the type of reading-writing task. There are three major kinds of reading comprehension- *literal*, *inferential* and *evaluative* [17]. There are four types of writing styles- *persuasive*, *narrative*, *expository* and *descriptive* [10]. The goal of the assignment can determine the style or combination of styles that a reader and writer may adopt. Despite these differences in reading and writing styles, writing tasks do have their commonalities across tasks- most involve reading, comprehending, and writing. With this view, we put up our case that creating a trace-based measurement protocol that can be used across multiple writing tasks can ease the pain of researchers who often have to conduct tedious controlled studies and manual coding to ascertain the validity of their trace data-based studies in their own context. Developing such a protocol can also help learning systems designers create universal learning environments which can support learners' self-regulation. Hence, we explore the possibility of generic trace-based measurement protocol that can measure SRL across multiple reading-writing tasks, and at the same time is able to identify the differences in self-regulation in each of these tasks.

In this doctoral project, we aim to investigate whether a trace-based measuring protocol designed, developed, and tested for one writing task can be used across multiple writing tasks. We also explore whether integrating multimodal data like eye-tracking with the existing log channel can improve the modeling of the learners.

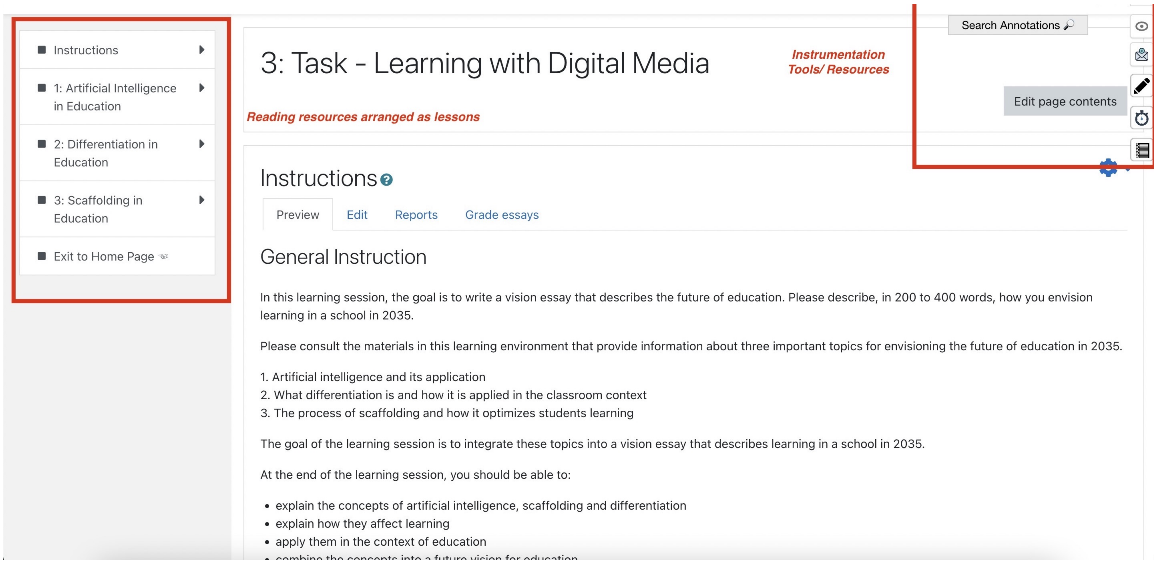
2. RESEARCH QUESTIONS

The following are the research questions that we aim to answer in this doctoral project:

1. **RQ1:** How do students' SRL strategies change when

D. Nath, D. Gasevic, and R. Rajendran. A trace-based generalized multimodal srl framework for reading-writing tasks. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 534–538, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115709


Instructions
1: Artificial Intelligence in Education
2: Differentiation in Education
3: Scaffolding in Education
Exit to Home Page
3: Task - Learning with Digital Media
Instrumentation Tools/ Resources
Reading resources arranged as lessons
Search Annotations
Edit page contents
Instructions
Preview
Edit
Reports
Grade essays

General Instruction

In this learning session, the goal is to write a vision essay that describes the future of education. Please describe, in 200 to 400 words, how you envision learning in a school in 2035.

Please consult the materials in this learning environment that provide information about three important topics for envisioning the future of education in 2035.

1. Artificial intelligence and its application
2. What differentiation is and how it is applied in the classroom context
3. The process of scaffolding and how it optimizes students learning

The goal of the learning session is to integrate these topics into a vision essay that describes learning in a school in 2035.

At the end of the learning session, you should be able to:

- explain the concepts of artificial intelligence, scaffolding and differentiation
- explain how they affect learning
- apply them in the context of education
- combine the concepts into a future vision for education

Figure 1: The Learning Environment

they engage in various reading-writing tasks with different goals?

2. **RQ2:** Do data from multiple sensors (like logs + eye-gaze) improve the detection of SRL strategies in learners, as compared to a single channel (i.e., logs)?

3. **RQ3:** Do prediction models trained on task-independent reading-writing multimodal data (data combined from multiple tasks) perform equivalently as for that in a specific reading-writing task?

3. THEORETICAL FRAMEWORK

Self-Regulated Learning (SRL) is a theoretical umbrella that encompasses cognitive, metacognitive, behavioural, and affective aspects of learning [14]. While different theoretical models have large commonalities between them as they try to capture alternate views of the same process, there exist subtle differences based on the aspects on which their central focus lies [14]. A large majority of these models view SRL as a cyclic process comprising three phases- Preparatory, Performance and Reflection. In the theoretical framework that we use, theoretically-grounded patterns of atomic user actions are mapped to higher-level SRL processes. Thus, different SRL processes have been operationalised using patterns of meaningful learner actions. A detailed description of the theoretical framework along with the exact list of patterns used to identify SRL processes is present in [5].

4. METHODS

Over the duration of the doctoral project, we aim to collect data from two (or more) reading-writing tasks, both with different content and different overall goals, and we aim to investigate them with a single trace-based SRL measurement framework. We will investigate whether the same framework is sufficient to capture the differences between the tasks, and what are the similarities as well. In the second year of PhD, we have focussed on collecting the data for one reading-writing task (specifically the one explained in section 4.2). A schematic diagram of our study design is represented in Fig 2.

4.1 Study Setup

The lab study is being conducted at the Department of Educational Technology, IIT Bombay. English is not the first language of the participants in the research study, but they have studied or are currently enrolled in institutions where English is the primary language of instruction. All the participants are college-going students from diverse streams or disciplines. The participants are a mix of undergraduates, post-graduates or PhD students.

As part of the data, we are collecting their software logs, the eye-tracking data of the students, their facial recordings and screen recordings. The eye-tracking data is being collected using Tobii Pro Nano screen-based eyetracker sampled at 60Hz. The data is exported using Tobii Pro SDK on Python which offers an open-source solution to export the raw data collected using Tobii eyetrackers.

4.2 Procedure

The study uses a pre-post test design which comprises of a 90 min reading-writing task the learners are required to go through a set of reading materials pertaining to three topics and compose a piece of writing. The three topics are: (1) artificial intelligence, (2) differentiation in the classroom and (3) scaffolding of learning. The goal of the task is to compose an essay that gives an overview of the state of education in the year 2035 within 400 words. The task has been designed in a way that prompts the learner to use SRL skills and tools like highlighter, notetaker in the learning environment.

4.3 Learning Environment

The task had been created in a Moodle-based learning environment, as shown in fig 1. The learning environment consists of a catalogue and navigation area which contains the list of reading materials and a way to navigate between

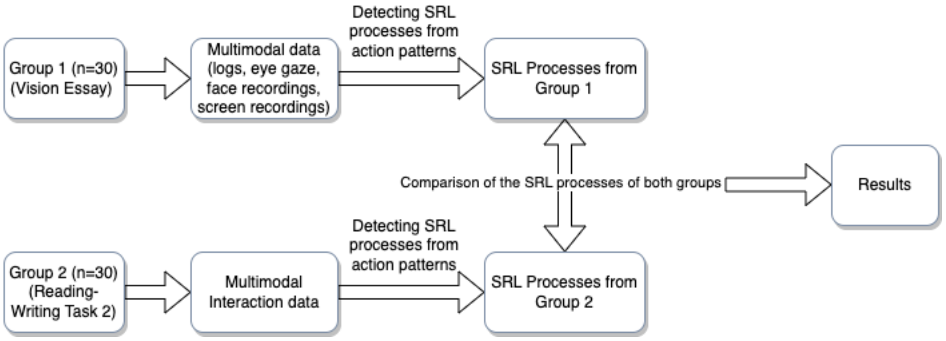

Group 1 (n=30) (Vision Essay)
Multimodal data (logs, eye gaze, face recordings, screen recordings)
Detecting SRL processes from action patterns
SRL Processes from Group 1
Comparison of the SRL processes of both groups
Results
Group 2 (n=30) (Reading-Writing Task 2)
Multimodal Interaction data
Detecting SRL processes from action patterns
SRL Processes from Group 2

Figure 2: The Study Design

them, and also to the general instructions and rubric of the task. There is a reading area in the centre which displays the contents of the selected reading material. The environment is integrated with tools like annotation, planner, timer tools. There is a writing window that can be opened and closed at any time for writing the essay. The planner tool helped the learners to plan how much time they are going to spend on each part of the task, and the timer tool displayed the time left for the task. The annotation tool, based on the open web annotation tool hypothes.is, allowed the learners to highlight, annotate and take notes, and they can search for highlights, tags and notes they created earlier as well. Within this learning environment, we collected learners' trace data that includes: 1) navigational log, which stored the time-stamps for all page visits; 2) mouse trace data, which stored mouse clicks on pages and mouse scrolls; 3) keyboard strokes.

4.4 Trace-based Measurement Protocol for Log Data

For converting the raw logs into theoretical SRL processes, we use a trace parser. The parser first converts the raw logs into meaningful learning actions like RELEVANT_READING, PLANNER, GENERAL_INSTRUCTION. These set of learning actions give rise to our *action library.* Specific theoretical patterns of these actions are then mapped to higher-level SRL processes. The entire list of these SRL processes then gives our *process library.* The entire process of parsing has been detailed in [5]. The theoretical SRL processes that we obtain in our learning task are also listed in Table 1. Each of these processes is coded by experts from 2-action or 3-action sequences. Our learning system also provides us with the duration spent on each of these patterns of actions (and hence the SRL processes). We can add the duration spent on each of these SRL processes separately during the 90 min learning period and also count their occurrences which gives us the metrics such as those represented in Table 1.

4.5 Data Processing and Feature Extraction from eye-gaze data

For cleaning and processing the raw eye gaze data, we will be following the steps outlined for the Tobii I-VT Fixation Filter [13]. The steps involve gap fill-in interpolation, eye selection, and noise reduction among other steps.

We will extract two main features- fixations and saccades and their derivatives from the eye-gaze data. For this purpose, we aim to use PyTrack [6], which is an open-source Python-based solution for analyzing eye-gaze data.

5. RESULTS

Table 1 represents the distribution of SRL processes within each category of SRL processes/subprocesses for 9 learners. The distribution is comparable to that presented in [5], where *Elaboration/Organisation*, *First Reading* and *Monitoring* emerged as the most prevalent SRL processes in the learners for the essay-writing task. A point to note is that the sample presented in [5] is from a population of learners whose first language is Dutch over 45 min of reading-writing, while the sample presented in this paper is from a population whose native language is not English over a period of 90 min.

Table 1: Distribution of SRL processes in the participants

Main Categories	Subcategories	Count	Duration (%)
Metacognition	Orientation	79	21.625
	Planning	10	0.375
	Monitoring	186	3.468
	Evaluation	17	0.574
Low Cognition	First Reading	267	36.974
	Re-reading	156	6.244
High Cognition	Elaboration/Organisation	349	30.739

6. FUTURE WORK

As introduced earlier, the objective of our task can determine the style or combination of styles that a reader and writer may adopt. To compare two examples, the vision essay in our learning task requires a learner to read and reflect on three readings- Artificial Intelligence in Education, Differentiation in Education and Scaffolding in Education and write a vision of education in 2035. The learner is expected to stay connected to the readings, but is also expected to combine them, go beyond what is there in the readings and imagine innovative scenarios in future where the information from these topics could be relevant. To contrast with this task, an argumentative task is a common form of academic writing where a learner is supposed to take a stance and make a for/against argument for a situation and back it up with evidences from the readings [4, 10]. Compared to the earlier vision essay, this task is rather restricted and the learner has to interpret the information, identify the relevant pieces of information from the readings, strictly adhere to facts and avoid misdirections in the text (if any) and put up a case for the argument. We hypothesize that such contrasting tasks can impact the self-regulatory behaviour of the learners, even while going through the same content.

For the research questions that we aim to address, we will continue our data collection. Once the data is collected for the current reading-writing task, we will change the task in terms of its goal and content, and collect data (most likely from a classroom course). This will allow us to have a substantial amount of data to answer our research questions in the ways described in brief below.

6.1 RQ1

To address RQ1, we aim to investigate the differences in the SRL strategies of learners depending on different reading-writing tasks using the following methods-

(a) Comparing the distribution of counts and duration spent by the learners in the SRL process categories for each of the tasks.

(b) Comparing aggregate process models of the learners for each of the tasks.

(c) Sequential Pattern Mining to reveal dominant action patterns in each of the task.

6.2 RQ2

To answer RQ2, we aim to combine log data (logs + mouse and keyboard interactions) and eye-gaze information. We plan to investigate whether sufficient attention was given each page of the content during each of the SRL process, and filter out the pages based on whether adequate eye-gaze were pointed to them.

6.3 RQ3

RQ3 involves a problem of prediction, which involves the prediction of the SRL process of the learner based on the logs and eye-gaze data of the students. The problem can be taken up either as a classification problem of predicting the SRL process from the data or predicting the next SRL process of the learner based on their current SRL state. We will train and test independently for each task, and compare the performance of our model when trained and tested for all tasks combined.

Prior to combining them, we will abstract features from the channels (features like count of mouse clicks, scrolls and count of fixations and saccades in AOIs from eye-gaze data).

7. CONCLUSION

The doctoral project focuses on investigating whether a single SRL measurement framework can be generalized for multiple reading-writing tasks. The outcome will provide evidence for the applicability of SRL measurement frameworks for multiple tasks and hopefully, it will prompt more research toward building generic SRL models at least for a certain set of tasks that have commonalities between them. The SRL measurement frameworks are at this point very contextual and restricted in nature.

The multimodal aspect of the project also aims to investigate whether additional data channels can reveal more information about the nature of self-regulation in learners. We will explore whether the eye-gaze channel can inform the log data channel better, or vice versa.

We have so far collected data for 16 participants, and have presented a summary of the results of 9 participants after consideration of the quality of the data. Going ahead we aim to collect more data from participants engaged in this task, and also collect data from learners in newer reading-writing tasks with different content. Then we will be ready to answer our RQs in ways described in the last section.

The approach is not without its limitations. The multimodal aspect of the project (especially RQ2) is very investigative in nature, and the methods will depend on the researcher. How to fuse the data channels, which exact features to select, and how to ensure its explainability is yet to be decided and are challenges on their own. The data that we have collected so far has been collected in a controlled lab environment, and real-world data might not be as clean as ours. We aim to collect data for our further reading-writing tasks from a real-world classroom, and ensuring the quality of the data and choosing appropriate technological solutions for multimodal data collection are other challenges. We also need to ensure that the content for our further new reading-writing tasks is comparable in terms of their complexity to the current reading-writing task.

The applicability of the research can be diverse and can go beyond just the measurement of SRL in reading-writing tasks. Although our major focus is on correct and valid measurement, appropriate measurement can be used for scaffolding learners' self-regulation which is an area that has gained momentum in recent years. Prediction models that we aim to investigate can help in scaffolding further by telling researchers which SRL processes the learner is going to enter next at any instant of time, in realtime. This information can be used to personalize the scaffolding process. Although at this point we only aim to work with logs and eyetracking data channels, more data channels like physiological sensors (skin conductance, heart rate) and facial expressions could be integrated to reveal more information about SRL [3].

8. ADVICE SOUGHT

The answer to the following questions will greatly help in ensuring that my research progresses on the correct path:

1. What are the best methods for comparing event-based processes? (other than sequential pattern mining, process models and statistical differences of event occurrences)?

2. The events in an activity such as the learner actions in our task occur at uneven intervals. Is there a possibility of using classic temporal prediction models in such cases?

3. How to combine data from multimodal channels while still keeping the temporal nature of the process intact, especially when the sampling rates of the data channels are uneven and one data channel (log data) is not even periodic in nature?

4. Are webcam eyegaze detection comparable to screen-based eye trackers when detecting fixations within an AOI?

9. REFERENCES

[1] S. Abadikhah, Z. Aliyan, and S. H. Talebi. Efl students' attitudes towards self-regulated learning strategies in academic writing. *Issues in Educational Research*, 28:1–17, 01 2018.

[2] M. Bernacki, J. Byrnes, and J. Cromley. The effects of achievement goals and self-regulated learning behaviors on reading comprehension in technology-enhanced learning environments. *Contemporary Educational Psychology*, 37:148–161, 04 2012.

[3] E. Cloude, R. Azevedo, P. Winne, G. Biswas, and E. Jang. System design for using multimodal trace data in modeling self-regulated learning. *Frontiers in Education*, 7:928632, 08 2022.

[4] ETS. Gre® general test: Analytical writing question types.

[5] Y. Fan, J. van der Graaf, L. Lim, M. Raković, S. Singh, J. Kilgour, J. Moore, I. Molenaar, M. Bannert, and D. Gašević. Towards investigating the validity of measurement of self-regulated learning based on trace data. *Metacognition and Learning*, May 2022.

[6] U. Ghose, A. S., W. Boyce, H. Xu, and E. Chng. Pytrack: An end-to-end analysis toolkit for eye tracking. *Behavior Research Methods*, 52, 03 2020.

[7] A. Hadwin, J. Nesbit, D. Jamieson-Noel, J. Code, and P. Winne. Examining trace data to explore self-regulated learning. metacognition & learning, 2, 107-124. *Metacognition and Learning*, 2:107–124, 12 2007.

[8] L. A. Hammann. Self-regulation in academic writing tasks. 2005.

[9] D. Jamieson-Noel and P. Winne. Comparing self-reports to traces of studying behavior as representations of students' studying and achievement. *Zeitschrift Fur Padagogische Psychologie - Z PADAGOG PSYCHOL*, 17:159–171, 11 2003.

[10] R. JEFFREY. OPEN OREGON EDUCATIONAL, 2018.

[11] P. K., S. T., and K. R. Development of computer-based learning system for learning behavior analytics. *Indonesian Journal of Electrical Engineering and Computer Science*, 25(1):460 – 473, 2022. Cited by: 0; All Open Access, Gold Open Access.

[12] R. Nitta and K. Baba. *Self-regulation in the evolution of the ideal L2 self: A complex dynamic systems approach to the L2 motivational self system*, pages 367–396. 01 2015.

[13] A. Olsen. The tobii I-VT fixation filter- algorithm description, Mar 2012.

[14] E. Panadero. A review of self-regulated learning: Six models and four directions for research. *Frontiers in Psychology*, 8, 2017.

[15] J. Saint, A. Whitelock-Wainwright, D. Gašević, and A. Pardo. Trace-srl: A framework for analysis of microlevel processes of self-regulated learning from trace data. *IEEE Transactions on Learning Technologies*, 13(4):861–877, 2020.

[16] N. Srivastava, Y. Fan, M. Rakovic, S. Singh, J. Jovanovic, J. van der Graaf, L. Lim, S. Surendrannair, J. Kilgour, I. Molenaar, M. Bannert, J. Moore, and D. Gasevic. Effects of internal and external conditions on strategies of self-regulated learning: A learning analytics study. In *LAK22: 12th International Learning Analytics and Knowledge Conference*, LAK22, page 392–403, New York, NY, USA, 2022. Association for Computing Machinery.

[17] D. o. E. Victoria State Government. Comprehension.

[18] P. Winne. Improving measurements of self-regulated learning. *Educational Psychologist*, 45:267–276, 10 2010.

[19] P. Winne. Construct and consequential validity for learning analytics based on trace data. *Computers in Human Behavior*, 112:106457, 06 2020.

Analyzing Team Cognition and Combined Efficacy In Makerspaces Using Multimodal Data

Nisumba Soodhani K
Indian Institute of Technology Bombay
nisumba@gmail.com

ABSTRACT

Makerspace has been growing as a major phenomenon since 2005. Learners' participation in makerspaces has proved useful in terms of their cognitive, affective and psychomotor outcomes. Many studies have reported on improved outcomes because of makerspaces, but how the learning process actually occurs is not clearly known. One reason for this is the makerspace setting itself which poses challenges for data collection, as makerspaces generally involve teams coming together and creating something. Capturing team dynamics in a real-time setting where mobility is hugely a part of it poses difficulty in multimodal data collection. To overcome the above-mentioned challenges and to understand the learning process in a makerspace, this thesis proposes multimodal data collection in a makerspace using a camera and eye tracker. Data will also be collected through surveys and interviews to understand team cognition, combined efficacy, and interests. Patterns will be identified and triangulated will inform us of the learner model and the learning process occurring in the makerspaces

Keywords

Makerspaces, multimodal data, team cognition, combined efficacy, self-efficacy, interest

1. INTRODUCTION

An increasing number of individuals are participating in the making of items in their daily lives and seeking ways to share their methods and artifacts with others through both physical and digital platforms [10]. The various learning theories associated with the maker movement are Seymour Papert's "constructionism", Jean Piaget's "constructivism", and John Dewey's idea of "learning by doing". Understanding these theories helps in designing and analyzing opportunities for learners to participate in makerspaces, create personalized projects and products that are meant to connect to students' own lived experiences demonstrate authenticity, and structure activities for enhancing teamwork and collaboration [4]. Literature signals strong links between interdisciplinary STEM (Science, Technology, Engineering, and Mathematics) and making, particularly the skills and capabilities utilized in projects, and opportunities to develop and apply STEM knowledge [9]. The National Research Council of the USA has recently identified makerspaces as learning environments with the potential for helping students to learn science and engineering concepts through investigation and design [5].

The relatively recent rise of the Maker Movement is a direct result of the widespread availability of low-cost digital fabrication technologies, the development of the Internet as a tool for sharing information, and an increase in media (e.g., Make magazine) and events (e.g., Maker Faires—community gatherings celebrating the Maker Movement) related to making [14]. In makerspaces collaboration is evident and the complexity of design problems requires that makers from different fields come together also a variety of scaffolds should be available to them to solve the problem. Organizations turn to teams in today's complicated and dynamic work environment to solve issues quickly and effectively. Teams-based organizational structures promote productivity, innovation, and other crucial organizational outcomes across industries [12].

The findings of the research also support the notion that makerspaces can aid in the development of a wide range of twenty-first-century skills [8]. Twenty-first-century skills (for example, collaboration, problem-solving, and digital citizenship) are a broad set of competencies that, when combined, indicate that individuals are prepared to be productive members of the workforce [13]. Research has been done to establish that there is some cognitive, affective and psychomotor gain, but limited research has examined how these skills and knowledge are developed. Even in those studies, qualitative methods such as observation, interviews, and self-reported surveys are heavily used. This thesis aims to address this gap by using multimodal data collection to understand the process of team cognition and also the role of self-efficacy and interest in it.

2. BACKGROUND

The recent developments in physiological sensing techniques technologies such as eye-tracker, EEG, wrist bands, etc., open ways to collect data in other modalities rather than focusing only on self-reports or questionnaires to understand the process of learning. They also have advantages such as

N. S. K. Analyzing team cognition and combined efficacy in makerspaces using multimodal data. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 539–542, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115683

less labour-intensive data collection over longer periods, allowing for the measurement of team cognition in real-world task contexts as opposed to simulated ones. Data can also be collected and analyzed in real-time which is also scalable. Data from one channel may not be enough to capture knowledge sharing, especially in a group setting where the focus is on team cognition and the combined efficacy of the team. Hence, there is a need to use data from multiple sensors. Multimodal analytics in makerspaces refers to the use of multiple types of data, or modalities, to gain a more comprehensive understanding of learners' activities and behaviours. This might include data from video cameras, sensor logs, and other forms of digital tracking, as well as more qualitative data such as interviews, surveys, and observations. By using multiple types of data, multimodal analytics can provide a more holistic view of the learners' experience in the makerspace and can help to identify patterns and trends that may not be visible when using only one type of data [2].

For example, sensor logs can provide data on the frequency and duration of use of different tools and resources, while video cameras can capture more detailed information on how learners are using those tools and resources. Interviews and observations can provide insight into learners' motivations, goals, and perceptions of their experiences in the makerspace [6]. By integrating these different types of data, multimodal analytics can help to identify patterns and trends that may not be visible when using only one type of data. It is important to note that multimodal analytics also involves a combination of quantitative and qualitative data [2]. This allows researchers to gain a deeper understanding of the learners' experiences in the makerspace and to identify patterns that may not be visible when using only quantitative data. For a better understanding of how knowledge is shared among team members and applied to solve problems, more research is required in the areas of team cognition, combined efficacy, and interest. Multimodal data analytics also has its own challenges, such as difficulty in temporally aligning data sources with different sampling rates and determining the amount of data to be sampled. Other challenges include the fusion of features from one modality to another for classification tasks, co-learning between modalities, and the generation of new features from one modality to another. Addressing these challenges is crucial for the effective analysis of multimodal data.

3. THEORETICAL FRAMEWORK

The suitable theoretical framework for understanding interests, beliefs, attitudes, and self-efficacy in makerspaces is the Self Determination Theory (SDT) developed by [7]. SDT is a framework that explains how individuals engage in activities and how that engagement is related to well-being and motivation. SDT suggests that individuals have innate psychological needs for autonomy, competence, and relatedness and that when these needs are met, individuals are more likely to engage in activities that are self-determined, intrinsically motivated, and lead to well-being. In the context of makerspaces, individuals who feel autonomous in their decision-making and have a sense of competence in their abilities to create and innovate will be more likely to engage in making activities.

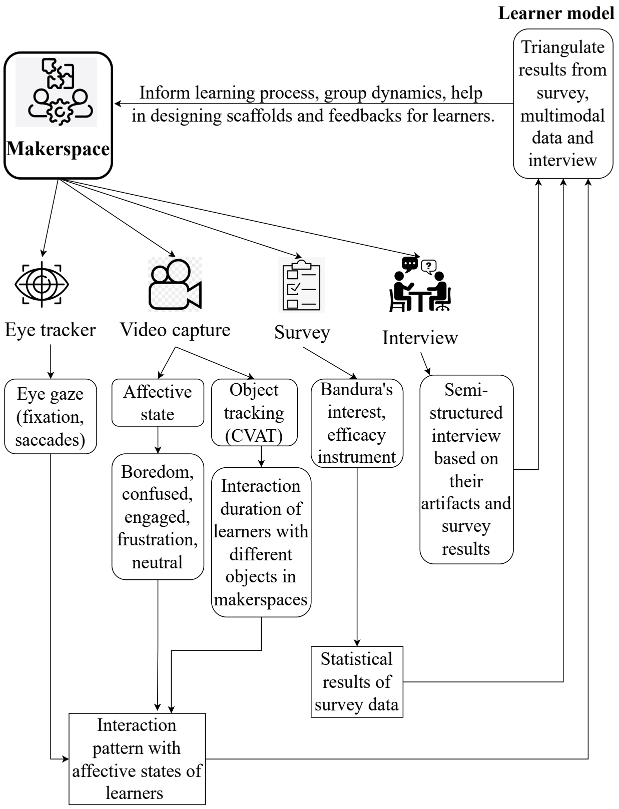

Learner model
Triangulate results from survey, multimodal data and interview
Inform learning process, group dynamics, help in designing scaffolds and feedbacks for learners.
Makerspace
Eye tracker
Video capture
Survey
Interview
Eye gaze (fixation, saccades)
Affective state
Object tracking (CVAT)
Bandura's interest, efficacy instrument
Semi-structured interview based on their artifacts and survey results
Boredom, confused, engaged, frustration, neutral
Interaction duration of learners with different objects in makerspaces
Statistical results of survey data
Interaction pattern with affective states of learners

Figure 1: Overview of the proposed method.

Interest, belief, and attitudes are also important factors in SDT. Interest is an intrinsic motivation for engaging in an activity, and beliefs and attitudes can influence an individual's perception of competence and autonomy in the activity. For example, if an individual holds a belief that they are not creative or do not have the necessary skills to participate in making activities, they may be less likely to engage in these activities [15]. Self-efficacy, or an individual's belief in their ability to perform a specific task, is also important in SDT. In makerspaces, individuals who have a high level of self-efficacy in their making abilities will be more likely to engage in making activities and persist in the face of challenges [15]. Overall, SDT provides a theoretical framework for understanding the factors that influence individuals' engagement in making activities in makerspaces, and how these factors are related to well-being and motivation [11].

4. RESEARCH OBJECTIVES

From the previous sections, we established that there is a need to investigate the interplay between various factors that influence students' interest, identity, and self-efficacy (beliefs in one's capabilities to organize and execute the courses of action required to produce given attainments) when they work together to solve problems collaboratively. Additionally, it is necessary to examine how each of these influences team cognition. To do this, I intend to take advantage of the

Table 1: Data Sources

Data type	Data source	Data
Qualitative data	Interviews and observation	They can provide valuable information about the resources and support provided in the makerspace, as well as the ways in which students are using the space and the impact it is having on their learning.
Quantitative data	Bandura's self-efficacy, interest survey	Self-report on individual self-efficacy, combined efficacy, and interest.
	Camera	Detect facial expressions. The video can also be used for object tracking and analyzing the amount the time spent by a learner interacting with different materials in the makerspaces.
	Eye tracker	Track the student's gaze and infer their level of engagement or interest in the task.

introduction of new technical tools like wearables and other covert measurement methods, which will provide us with the chance to advance our understanding of the science behind team cognition. Because of these technical developments, it is now possible for academics to evaluate data streams that are far larger than they have ever been. Additionally, when combined with conventional metrics, these tools can give additional context for comprehending the level of cognition among teams. Collectively, these efforts will

1. Inform researchers about how knowledge is shared in team cognition, interest, and combined efficacy's role in it.
2. Understand and model participants' learning processes.
3. Inform designers and developers to provide scaffolding and feedback.

5. RESEARCH METHODOLOGY

5.1 Preliminary study

The preliminary study was conducted with fourteen participants who were introduced to digital making – TinkerCAD and Scratch. All participants identified themselves as female. Participants discussed the socio-environmental and economic issues with their peers and came up with a critical making design plan to tackle the identified problem. The plan or ideas submitted by them were the artifacts and their responses to the survey questionnaire focusing on self-efficacy, beliefs, and attitudes were the primary data sources. This questionnaire was adapted from Bandura's self-efficacy scale [3]. This survey was administered after the workshop. Their artifacts were analyzed using content analysis and the survey results were mapped to the artifacts. This pilot study helped in understanding the self-efficacy and interest of first-time makers. The quality of artifacts and the statistical results of surveys had a correlation. Participants who reported high efficacy had better artifacts in terms of their actionable plan.

5.2 Participants and Data Collection

The future study will be conducted primarily amongst undergraduate program students as individuals at this level are young adults and usually are at the starting point of shaping their lives based on their interests and have certain autonomy to do so. The data that will be collected and the data sources are mentioned in Table 1. Data collected from self-reported surveys and interviews will be mapped with the patterns and findings from multimodal data analytics. In order to understand the learner's behaviour a quantitative approach will be used which will employ machine learning.

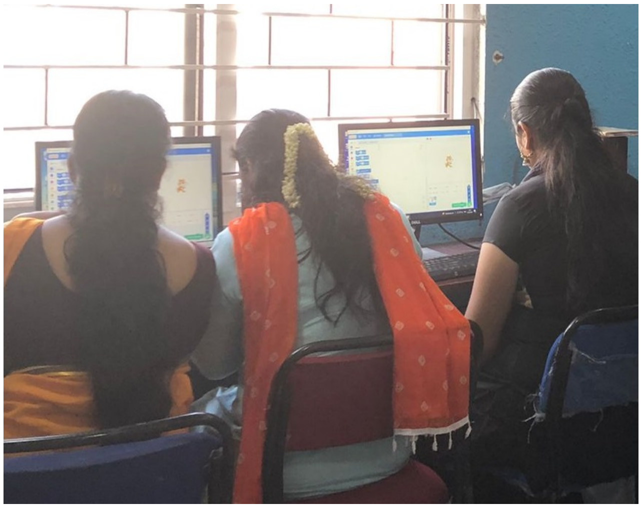


Figure 2: Participants working on Scratch.

5.3 Data Analysis

The video can be used for object tracking with CVAT, which can record the duration of interaction with a specific object [1]. This information, combined with eye gaze data, can provide insight into a task's level of engagement and interest. The combined efficacy and interest questionnaire survey results can be statistically analyzed to provide additional data. To identify patterns and understand the meaning of these patterns in the context of the data, the interview data can be coded and analyzed using grounded theory. The patterns that emerge from the multimodal data can be used to triangulate the qualitative analysis findings, providing a more comprehensive understanding of the learning process in the makerspaces. Using this method of study and analysis will help in coming up with a more detailed and nuanced understanding of the process. The triangulation with survey results and interview data will help us in explaining the role of self-efficacy and combined efficacy in social learning environments like makerspaces.

6. ONGOING AND FUTURE WORK

The pilot study is completed and the next step in the research process is to conduct data collection and analysis of the primary study. The data collection should involve gathering qualitative and multi-modal data from the makerspaces, such as observations, interviews, and documents. This data should then be analyzed in order to address the research question and answer the study's objectives. One potential challenge when analyzing multimodal data is finding a way to effectively combine and analyze data from multiple modalities, such as eye gaze, and video. This may involve using specialized software or techniques and may require consulting with experts in the field of multimodal data analysis.

7. CONCLUSION

Makerspaces are defined by groups of people getting together to create something in real-time, which requires a lot of movement, making data collecting challenging. This thesis proposes the use of multimodal data gathering to better understand the learning process in makerspaces. While the advantages of makerspaces for learners have been well acknowledged, the specifics of how learning takes place in this context have remained unknown due to data-gathering issues. In addition to questionnaires and interviews, the suggested use of a camera and eye tracker attempts to overcome these limitations and give a more thorough knowledge of the cognitive, emotional, and psychomotor effects of involvement in makerspaces. The identified patterns will be triangulated to inform a learner model and shed light on the learning process occurring in makerspaces. This will provide insight into group dynamics, learning processes, and help designers in scaffolding and providing feedback for learners.

8. REFERENCES

[1] T. Ashwin and R. M. R. Guddeti. Affective database for e-learning and classroom environments using indian students' faces, hand gestures and body postures. *Future Generation Computer Systems*, 108:334–348, 2020.

[2] R. Baker and G. Siemens. Learning analytics and educational data mining. *Cambridge handbook of the leaning sciences (2nd edn). Cambridge University Press: New York, NY*, pages 253–272, 2014.

[3] A. Bandura et al. Guide for constructing self-efficacy scales. *Self-efficacy beliefs of adolescents*, 5(1):307–337, 2006.

[4] R. A. Brown and A. Antink-Meyer. Makerspaces in informal settings. *Educational Technology*, pages 75–77, 2017.

[5] N. R. Council et al. *STEM integration in K-12 education: Status, prospects, and an agenda for research*. National Academies Press, 2014.

[6] N. Dabbagh and A. Kitsantas. Personal learning environments, social media, and self-regulated learning: A natural formula for connecting formal and informal learning. *The Internet and higher education*, 15(1):3–8, 2012.

[7] E. L. Deci and R. M. Ryan. Self-determination theory. 2012.

[8] M. Doorman, R. Bos, D. de Haan, V. Jonker, A. Mol, and M. Wijers. Making and implementing a mathematics day challenge as a makerspace for teams of students. *International Journal of Science and Mathematics Education*, 17:149–165, 2019.

[9] G. Falloon, A. Forbes, M. Stevenson, M. Bower, and M. Hatzigianni. Stem in the making? investigating stem learning in junior school makerspaces. *Research in Science Education*, pages 1–27, 2020.

[10] E. R. Halverson and K. Sheridan. The maker movement in education. *Harvard educational review*, 84(4):495–504, 2014.

[11] S.-Y. Han, J. Yoo, H. Zo, and A. P. Ciganek. Understanding makerspace continuance: A self-determination perspective. *Telematics and Informatics*, 34(4):184–195, 2017.

[12] S. Khaleghzadegan, S. Kazi, and M. A. Rosen. Unobtrusive measurement of team cognition: A review and event-based approach to measurement design. *Contemporary Research*, pages 95–113, 2020.

[13] L. C. Larson and T. N. Miller. 21st century skills: Prepare students for the future. *Kappa Delta Pi Record*, 47(3):121–123, 2011.

[14] R. Rouse and A. G. Rouse. Taking the maker movement to school: A systematic review of prek-12 school-based makerspace research. *Educational Research Review*, 35:100413, 2022.

[15] V. W. Vongkulluksn, A. M. Matewos, G. M. Sinatra, and J. A. Marsh. Motivational factors in makerspaces: a mixed methods study of elementary school students' situational interest, self-efficacy, and achievement emotions. *International journal of STEM education*, 5:1–19, 2018.

Exploring students' learning processes by logging and analyzing their interaction behavior in a Virtual Reality learning environment

Antony Prakash
Indian Institute of Technology Bombay
antony80004@gmail.com

ABSTRACT

Educational researchers have done remarkable work in analyzing the impact of VR on education and measuring the learners' experience, engagement, motivation, etc of using VR in education. Most of the studies conducted reveal that VR in education has a positive impact as the learners immersively experience the Virtual Reality Learning Environment (VRLE) and interact with the virtual objects in the first-person perspective. Certain experiments conducted with VR also claim that using VR in education increases the presence but decreases learning due to an overload of extraneous cognitive load. Due to the contradictory claims made by different authors on the use of VR in education, it has become important to understand the learning processes happening while using VR. The existing studies conducted to understand the learning processes in VR have considered the cognitive factors, and affective factors leading to the learning outcomes. Those studies have used the data collected from pre-tests, post-tests, self-reported questionnaires, interviews, surveys, physiological devices, and human observers. However, no study has considered the data related to the behavior of the learners due to interacting with VRLE to understand the learning processes. This is due to the non-existence of an efficient data collection mechanism that is able to log all the interaction behavior of the learners. Hence, we developed a data collection mechanism that is able to log automatically all the interaction behavior of the learners in real time along with timestamps. We also conducted a study with 14 participants by deploying the developed data collection mechanism in a VRLE. The purpose of this doctoral thesis is to understand the learning processes happening in VRLE from the lens of the interaction behavior of the learners. The analysis done on the data collected can also be further used to predict the learners' performance based on their interaction behavior.

Keywords

interaction behavioral data, behavioral pattern, pattern mining, modeling learners, personalization, VRLE

A. Prakash. Exploring students' learning processes by logging and analyzing their interaction behavior in a virtual reality learning environment. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 543–547, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115659

1. INTRODUCTION AND RELATED WORK

Virtual Reality (VR), is the technology that can make the users experience a 3D virtual world by immersing in it and interacting with the virtual objects present there in a first-person perspective similar to the real world. Virtual Reality (VR) technology, due to its unique characteristics of immersion, interaction, and imagination [15, 14, 10] has found its application in various domains such as the Automotive industry, Military, Healthcare, Sports domain, etc including the education domain. In education, the learners use VR to acquire knowledge and skills on the learning contents that involve 1) invisible phenomena such as electricity, magnetism, etc [9], 2) microscopic concepts such as DNA [12], a human artillery system [8], etc, and 3) dangerous and hazardous procedures such as fire fighting, welding, etc [11]. As the application of VR in education is increasing, the number of research being done on VR in education has also seen an exponential increase in the last decade. The experiments conducted in the research so far have used the data collected from 1) pre-tests and post-tests to measure the impact of VR on learning, 2) self-reported questionnaires, interviews, and surveys [10] to measure the user experience, engagement, and usability of the VR systems and to compare VR-aided and VR-non-aided learning systems [1], 3) devices such as i) physiological sensors to assess the affective state of the learners while performing the tasks [3], ii) eye trackers to assess the learners' intended area of interest [13] iii) body trackers to adapt the size of the virtual objects with respect to the size of the users, and iv) orientation of the head-mounted displays (HMD) and handheld controllers (HHC) to assess the response time, and 4) human observers to understand the behavior and procedural performance of the learners [8]. The learning outcomes measured in the existing studies using the existing data collection mechanism are 1) cognitive skills (knowledge acquisition, knowledge retention, and knowledge transfer), 2) affective skills (motivation, satisfaction, etc), and 3) procedural skills (sequential execution, duration of completion) [10]. In spite of a lot of work being done on measuring the learning outcomes, there is little work done to understand the learning processes to know about how the learners learn in Virtual Reality learning environment (VRLE). The limited works done to understand the learning processes too have considered the cognitive factors, and affective factors [4, 2, 7]. However, the procedural skills constituting one of the learning outcomes are not considered in understanding the learning processes. This could be due to the fact that procedural skills are measured using the data related to the behavior of the learners provided by

human observers [8]. However, the data provided by human observers can get biased and also need to satisfy interrater reliability tests [7]. Hence there is no efficient mechanism to collect behavioral data of the learners while they interact with VRLE. Moreover, the studies conducted to understand the learning processes have been done on a desktop VR rather than in an immersive VR learning environment [6]. The desktop VR system uses a mouse and keyboard for interaction. Whereas, in immersive VR systems, interactions happen through the buttons of hand-held controllers (HHC). As there is no mechanism existing to collect interaction behavioral data (IBD) in immersive VRLE, there has been no research done to understand learning processes in immersive VR systems by considering the learners' behavior. Therefore, this research thesis aims to explore learning processes using IBD collected in a VR learning environment. To reach this aim, the research thesis will follow three main phases. First, the development of a specialized IBD collection mechanism and its deployment in an immersive VRLE. Second, the collection of interaction behavioral data from studies conducted in the VRLE to explore the learning processes. Third, the development of a VR-based adaptive tutoring system that can provide personalized adaptive feedback, scaffolds, and VR learning content to the learners based on their interaction behavior. The contributions of the research project are:
1) Measuring the impact of VR on learning the subject area of electronics engineering as VR studies in electronics engineering are limited.
2) Deployment of the developed IBD collection mechanism as there was no system existing to log the learners' behavior in real-time.
3) An approach to predict the learning using the IBD and the performance of the learners by fitting them with a regression model.
4) An approach to model the learners' behavior using the IBD logged and process mining techniques

2. CURRENT RESEARCH PROGRESS

The important works done in our research so far are 1. We adopted and improvised MaroonVR [9], an open-sourced VRLE used to learn the physics concepts of electromagnetic induction. Electromagnetic induction is a phenomenon in which the electromotive force (emf) is induced when a magnet is moved through a closed loop coil. We used two scenes of MaroonVR viz Faraday's law scene and the falling coil experiment scene. The improvisations are done to the VRLE by enhancing interactions in the Faraday's law scene, converting the simulated falling coil experiment scene into an interactive scene, and including an embodiment-integrated learning scene in the environment. In the embodiment scene, the learners take the perspective of the magnet and through their walking action the emf gets induced rather than due to dragging the magnet in the other two scenes. The modifications were made to MaroonVR to make the VRLE more suitable for experiential learning [11] and embodied learning [5] to happen. The learners can feel the haptic vibration when the emf gets induced due to the magnet dragging in and out of the coil. The emf also gets plotted in real-time in the virtual graph present in the VRLE. 2. We developed an IBD collection mechanism that is able to log all the interaction behavior of the learners in VRLE in real time along with the time stamp. The process involved in the development of the IBD collection mechanism contains two steps viz. i) creation of a report folder to save the data file in .csv, and ii) appending the log data into the created .csv report file as shown in Figure 1. More details about the IBD collected is discussed in section 3. 3. We deployed the IBD collection mechanism in MaroonVR and conducted a study with 14 engineering undergraduate students and collected the IBD.

3. INTERACTION BEHAVIORAL DATA

We use Oculus Quest 2 from Meta, an immersive VR system in which the learners view the VRLE using a head-mounted display (HMD) and interact with the objects present in it using the buttons present in the hand-held controllers (HHC). The interactive actions performed on the virtual objects of VRLE by various buttons present in HHC constitute interaction behavioral data (IBD) which are discussed in the following sub-sections.

3.1 Interaction Behavioral Data Logger

The IBD collection mechanism deployed in the VRLE collects information about the interaction made through the HHC buttons, the virtual objects with which the interactions happen, and the timestamp. As we employ the Unity game engine to modify and program the interactions in the VRLE, we wrote a *c#* code to create a folder in the desktop computer (to which HMD is tethered) in which all the interaction behavior data can be logged in a .csv file. We use the VR Tool Kit (VRTK) package of Unity to collect all the actions done on HHC buttons. We wrote another *c#* code in Unity to append the .csv file with information about the virtual objects with which the interaction has happened. The code is written so that the timestamp gets recorded in a separate column corresponding to the respective rows to which new data is appended. The IBD collection mechanism is shown in Figure 1.

3.2 Buttons and Interactions

The buttons of the HHCs that involve in the interactions are the trigger buttons, grip buttons, thumbstick, primary buttons, and secondary buttons. The trigger buttons are generally used to interact with the virtual interfaces such as interface buttons, and interface sliders. In MaroonVR, the interface buttons that are used to change the number of turns in the coil, and to change the diameter of the area enclosed by the coil can be interacted with using the trigger button. The grip buttons are generally used to grab, drag, and drop/throw virtual objects. In MaroonVR, virtual objects such as virtual magnets, iron bars, and door handles can be interacted with using the grip button. Thumbstick present in the HHC can be used to teleport from one place to another place within the 3D world without actually moving in the real world. The primary button and the secondary button are used to switch to and switch back from the embodiment scene in MaroonVR respectively. The HHC buttons and the interactive actions performed by them are tabulated in Table 1.

3.3 Action Logs

The interactive actions performed using various buttons of HHC as mentioned in Table 1 are logged into the IBD logger. The IBD logger contains information such as HHC used (left

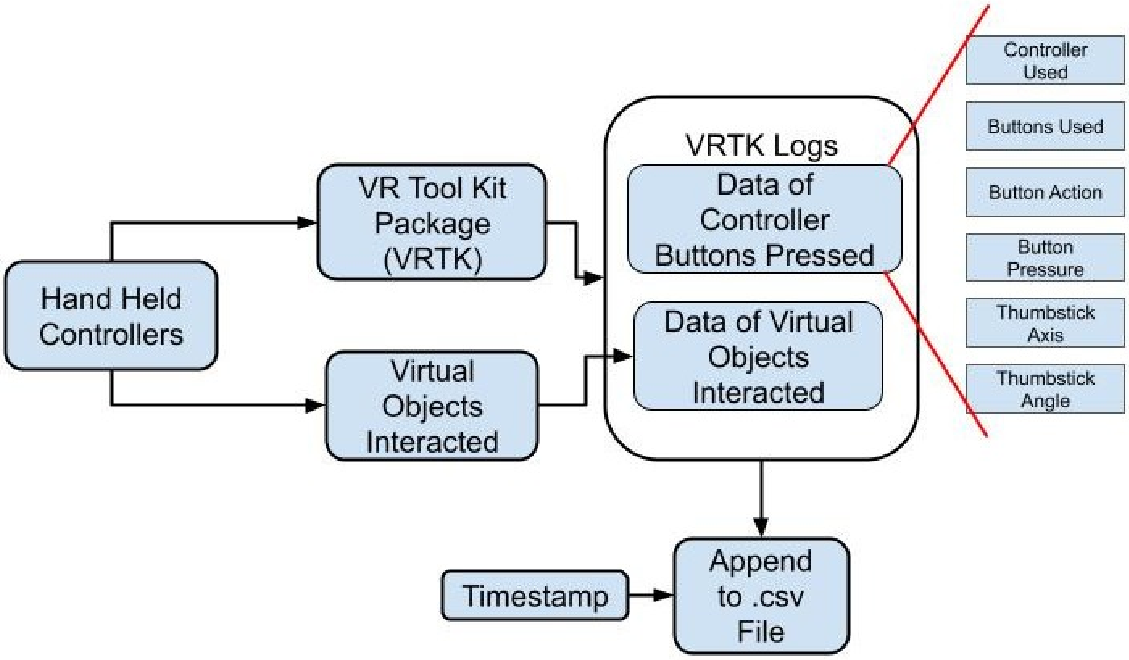


Figure 1: Interaction Behavioral Data Collection Mechanism in VR

Table 1: HHC Buttons and Interactive Actions

Controller Buttons	**Interactive Actions**	**Virtual Objects Interacted**
Trigger	Select, Unselect, and Change values	Virtual user interfaces such as virtual buttons to select between different coil turns, different coil diameters, play/pause buttons, and sliders
Grip	Grab, Drag, and Drop	Magnet, Iron bar, door handles, Perspective scene
Thumbstick	Teleport	-
Primary Button	Switch to embodiment scene	-
Secondary Button	Switch back from embodiment scene	-

or right), buttons of the HHC used (refer to Table 1), button actions (pressed, released, clicked, etc), button pressure (a value between 0 and 1), thumbstick axis (x and y coordinates), and thumbstick angle (angle range between 0° and 360°), objects interacted (refer to Table 1), and timestamp in different columns of the .csv report file. The actions done by the learners can be identified from the buttons of the HHC used, button actions, and objects interacted columns of the IBD logger. The action logs identified from the IBD logger such as reading instruction, coil interaction, magnet interaction, iron bar interaction, scene switching, and non-interactive actions are described in Table 2.

Table 2: Action Logs

Actions	**Description**
Reading instruction	Reading the instructions before starting the interactions
Coil interaction	Changing the parameters of the coil such as number of turns and diameter
Magnet interaction	Grabbing, dragging, and dropping of the magnet present in Faraday's law experiment lab, and falling coil experiment lab
Iron bar interaction	Grabbing, dragging, and dropping of the iron bar present in falling coil experiment lab
Scene switching	Moving from one scene to another scene among three different scenes
Non-interactive actions	The learners look and walk around in the VRLE rather than interacting with virtual objects

3.4 Experimentation

We conducted a study with 14 participants who are undergraduate engineering students from a non-electrical background. The data relating to the participants' self-efficacy and self-regulated learning were collected from the responses provided by the participants to the respective questionnaires. A pre-test was also conducted to assess the participants' prior knowledge of electromagnetic induction. The participants were then allowed to play a VR game known as "First Steps" for 15 minutes to get familiarize themselves with the VR system. Then they interacted with MaroonVR VRLE for 30 minutes and the data related to their interaction behavior gets collected automatically in the IBD logger. After the VR intervention, a post-test was conducted to assess the learning outcome. The participants then answered a series of self-reported questionnaires such as learners' experience (M=6.66, SD=0.47), and enjoyment (M=5.22, SD=0.48) on a 7-point Likert scale and VR engagement (M=4.21, SD=1.18) on a 5-point Likert scale. Also, no participant reported any kind of motion sickness or nausea during or after the VR intervention as the movement of the participants in the virtual world and real world is synchronized.

Table 3: Actions Frequency and Duration Calculated from the IBD collected in the study

Actions	Frequency (number of Occurrences)	Duration(in seconds)
Reading instruction	14	4160
Coil interaction	67	-
Magnet interaction	118	1397
Iron bar interaction	30	97
Scene switching	29	-
Non-interactive actions	-	17255

The details about the behavior of the participants with respect to actions, frequency of occurrences of the actions, and the duration of performing the respective actions during VR intervention are shown in Table 3. The entries in the table show the behavior of all 14 participants collectively. The action of coil interaction involves the interaction of varying the coil parameters such as turns, and diameter by clicking using the trigger button in HHC. Similarly, the action of scene switching also involves clicking the door handle to enter another scene. Hence, the number of occurrences of these actions is evaluated rather than the duration of occurrences. Whereas, for non-interactive actions such as walking and looking around the environment, duration is calculated rather than frequency.

4. LEARNING PROCESSES ANALYSIS

We have developed an IBD collection mechanism, deployed it in a VRLE, conducted a study with 14 participants, and identified the actions done by the learners from their interaction behavior. All through these processes we used experiential learning theory that has four components viz concrete experience, reflective observation, abstract conceptualization, and active experimentation [11]. The existing studies on the learning processes in VR have considered only the cognitive factors and affective factors [4] that are the aspects of the first three elements of experiential learning theory. Whereas, active experimentation which can be assessed by the interaction behavior is not considered in understanding the learning processes in VR. Therefore, we will extract the temporal and spatial features of the actions identified from the IBD collected and do pattern mining to find the behavioral pattern of the learners leading to higher performance. This would help us to find how learners learn in a VRLE from the lens of interaction behavior. We also propose that the results obtained would be further used to model the learners' proficiency. We also propose to develop an algorithm to provide personalized adaptive feedback, scaffolds, and learning content based on learners' interaction behavior and proficiency.

5. CONCLUSION

We conclude that IBD developed and deployed in the VRLE is able to log all the interactive actions performed by the learners. However, the experiment was conducted with a small sample of 14 participants. Hence, to establish the generalizability of the study the experiment needs to be conducted with a larger number. As the learners experiencing the VRLE are expected to see the virtual graph while they interact with the magnet, the information related to their seeing needs to be logged. However, the current data collection mechanism is unable to provide information related to the learners' seeing. Hence, further work is required to ensure that the learners see the intended area while they perform tasks in the VRLE and relevant information needs to be logged. We will use the logged IBD to explore the learning processes in VR. Also, in the current experiment, the learners interacted with the virtual objects and viewed the virtual graph for the corresponding changes in the voltage level. In the future, the VRLE will be modified to include task-based challenges like glowing electric bulbs having different wattage ratings by interacting with virtual objects. We also propose to develop a mechanism to provide personalized feedback, scaffolds, and VR learning content to the learners based on their behavior in VRLE.

During the doctoral consortium, we expect to recommend suggestions and feedback related to our current progress in our research. We specifically expect the recommendations on establishing the connection between the behavior of the learners and the learning outcome, and personalization of the learning system in the VR context.

6. REFERENCES

[1] P. Albus, A. Vogt, and T. Seufert. Signaling in virtual reality influences learning outcome and cognitive load. *Computers & Education*, 166:104154, 2021.

[2] B. Chavez and S. Bayona. Virtual reality in the learning process. In *World conference on information systems and technologies*, pages 1345–1356. Springer, 2018.

[3] Z. Feng, V. A. González, R. Amor, R. Lovreglio, and G. Cabrera-Guerrero. Immersive virtual reality serious games for evacuation training and research: A systematic literature review. *Computers & Education*, 127:252–266, 2018.

[4] G. Makransky and G. B. Petersen. Investigating the process of learning with desktop virtual reality: A structural equation modeling approach. *Computers & Education*, 134:15–30, 2019.

[5] G. Makransky and G. B. Petersen. The cognitive affective model of immersive learning (camil): A

theoretical research-based model of learning in immersive virtual reality. *Educational Psychology Review*, 33(3):937–958, 2021.

[6] G. Makransky, T. S. Terkildsen, and R. E. Mayer. Adding immersive virtual reality to a science lab simulation causes more presence but less learning. *Learning and instruction*, 60:225–236, 2019.

[7] E. Olmos-Raya, J. Ferreira-Cavalcanti, M. Contero, M. C. Castellanos, I. A. C. Giglioli, and M. Alcañiz. Mobile virtual reality as an educational platform: A pilot study on the impact of immersion and positive emotion induction in the learning process. *EURASIA Journal of Mathematics, Science and Technology Education*, 14(6):2045–2057, 2018.

[8] R. Pathan, R. Rajendran, and S. Murthy. Mechanism to capture learner's interaction in vr-based learning environment: design and application. *Smart Learning Environments*, 7(1):1–15, 2020.

[9] J. Pirker, M. Holly, I. Lesjak, J. Kopf, and C. Gütl. Maroonvr—an interactive and immersive virtual reality physics laboratory. In *Learning in a Digital World*, pages 213–238. Springer, 2019.

[10] J. Radianti, T. A. Majchrzak, J. Fromm, and I. Wohlgenannt. A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers & Education*, 147:103778, 2020.

[11] G. Sankaranarayanan, L. Wooley, D. Hogg, D. Dorozhkin, J. Olasky, S. Chauhan, J. W. Fleshman, S. De, D. Scott, and D. B. Jones. Immersive virtual reality-based training improves response in a simulated operating room fire scenario. *Surgical endoscopy*, 32(8):3439–3449, 2018.

[12] L. Sharma, R. Jin, B. Prabhakaran, and M. Gans. Learndna: an interactive vr application for learning dna structure. In *Proceedings of the 3rd International Workshop on Interactive and Spatial Computing*, pages 80–87, 2018.

[13] J. Wade, L. Zhang, D. Bian, J. Fan, A. Swanson, A. Weitlauf, M. Sarkar, Z. Warren, and N. Sarkar. A gaze-contingent adaptive virtual reality driving environment for intervention in individuals with autism spectrum disorders. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 6(1):1–23, 2016.

[14] L. Yang, J. Huang, T. Feng, W. Hong-An, and D. Guo-Zhong. Gesture interaction in virtual reality. *Virtual Reality & Intelligent Hardware*, 1(1):84–112, 2019.

[15] H. Zhang. Head-mounted display-based intuitive virtual reality training system for the mining industry. *International Journal of Mining Science and Technology*, 27(4):717–722, 2017.

Data Driven Online Training Program for Education Robotics Competition

Suprabha Jadhav[a*], Sridhar Iyer[a] & Kavi Arya[a]
[a]Indian Institute of Technology Bombay, India
*suprabhaj@iitb.ac.in

ABSTRACT

Educational Robotics (ER) is a field of study that aims to promote active learning and engage students with the use of artifacts. An IIT BOMBAY project, "e-Yantra" uses Project Based Learning (PBL) approach to train students to be able to solve real-world problems through an Educational Robotics (ER) competition through one of its initiatives titled "e-Yantra Robotics Competition (eYRC)". Students participate in the competition to gain skills, knowledge, and hands-on learning experience. But due to lack of thinking skills, exposure to different domains and other constraints like academic commitments, students find it difficult to compete and eventually drop out of the competition. To address this problem, this project focuses on designing a data driven training program that will prepare students to help gain skills required for Educational Robotics competition.

Keywords

Educational Robotics (ER), robotics competition, training program.

1. INTRODUCTION

Educational Robotics is a research field that positively impacts the students' learning experience by implementation of hands-on activities where robots play an important and active role [1]. Robotics activities can promote different learning outcomes such as problem solving, self-efficacy, computational thinking, creativity, motivation, and collaboration. Many robotics kits have been designed and developed for educational purposes that provide opportunities for students to explore, implement and receive feedback. To benefit students from a robotics competition, aspects such as design of competition, student training, mentor's scaffolding, and teaching pedagogies are important [2].

e-Yantra conducts an online annual Robotics Competition for students to implement solutions to the real-world problems on sectors like waste segregation, medicine delivery, road maintenance, soil monitoring etc. Competition comprises detailed problem statement, task documentation, self-paced video tutorials, robotics kits, discussion forum, live mentor interaction to help students to learn, compete, and resolve their queries [3].

Few of the popular international competitions such as World Robot Olympiad (WRO) – India [4], Micromouse, RoboGames, ABU Robocon [5], RoboCup (Robot Soccer World Cup), VEX Robotics Competition, Zero Robotics tournament, Robofest India, B.E.S.T Robotics Design Contest, Botball Educational Robotics Program, FIRST: Robotics Competition are designed for students of different age groups.

2. CURRENT AND PROPOSED WORK

The aim of my work is to create an online training program for undergraduate students participating in the e-Yantra Robotics Competition. To attain this, following are the goals:

Goal 1: Examine the need for an online training program using a data-driven approach.

Goal 2: Define the structure of the training program.

Goal 3: Determine the effectiveness of the training program.

To address Goal 1, I did a thorough analysis of 11 well-known tournaments. The investigation included determining the competition's purpose and categories, target audience, mode of conduct, training, resources provided (before, during, and after the competition), mentor participation, and role.

On the official website of competitions, information about the above factors was found but specific information about training and resources provided to students during competition and the role of mentors and other scaffolds made available was not found. On the other hand, competitions do provide some resources, notes, guides, rulebooks, certification courses for educators.

From the studies [8] and [9], it is evident that a major attrition rate is seen especially after the initial task i.e., Task 0. Major self-reported reasons include task difficulty, difficulty managing time, team coordination issues, beginners (participating for the first time), university exams clashing with competition task deadlines, participation in other events, and so on. To understand the issue further, data was collected in two ways:

A. Semi-Structured Interviews
B. Survey Form

A. Semi-Structured Interviews:

Interviews were conducted for students/teams of the ongoing competition edition 2022. It was not feasible to conduct interviews for all the participating teams on each theme given the number as highlighted in Table 1 below.

Table 1. e-Yantra robotics competition theme details

Theme name	No. of teams
Sentinel Drone (SD)	374
Functional RoadBot (FB)	376

S. Jadhav, S. Iyera, and K. A. D. D. O. T. P. for Education Robotics Competition. Data driven online training program for education robotics competition. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 548–551, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115641

Theme name	No. of teams
Swatchhta Bot (SB)	372
Delivery Bike (DB)	372
Krishi Bot (HB)	372
Pharma Bot (PB)	372
HolA Bot (HB)	372
Total	2610

Though eYRC is a collaborative competition, it was also important to know individual insights which might be missed in a team interview. Individual interviews were decided for the following two reasons:

- If there are ongoing team clashes, the student may share without being judged or feared by other team members
- For a particular question, if one student shares some insight, another team member might not take the effort of thinking and would end up saying the same thing.

To calculate the number of interviews considering different factors, the following was planned:

Table 2. Interview preparation details

Total themes	7	
Level	Low, Medium, High scorers	
Categories	Individual	Team
No. of teams to be interviewed/theme	1	1
Total members/team	2-4	2-4
Total no. of students to be interviewed	2-member team: 14 4-member team: 28	NA
Total no. of teams to be interviewed	NA	7
Total	14-28 students	Team

Random sampling was done to choose low, medium, and high scorer teams for 7 themes as shown in Table 3:

Table 3. Interview categories

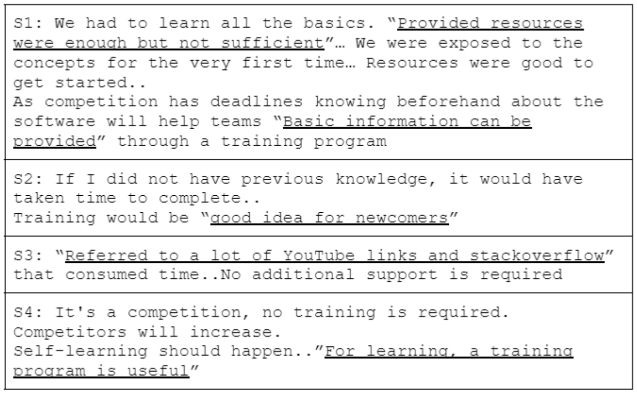
Score/Level	Individual	Team
Low	HB, PB	FB, SB, DB
Medium	SD, SB, DB	KB, PB
Top	KB, FB	SD, HB

The objective of the interview was to understand challenges faced by participants, resources availability, and need for additional training. After 11 interviews, I started getting similar responses so no more interviews were conducted. The interview details are stated in Table 4.

Table 4. Interview details

Interview Time	~ 40 mins each
Platform	Webex
Data collection	Audio and Video recording

Figure 1 and Figure 2 shows few instances from interview transcript analysis:



S1: We had to learn all the basics. "Provided resources were enough but not sufficient"… We were exposed to the concepts for the very first time… Resources were good to get started..
As competition has deadlines knowing beforehand about the software will help teams "Basic information can be provided" through a training program

S2: If I did not have previous knowledge, it would have taken time to complete..
Training would be "good idea for newcomers"

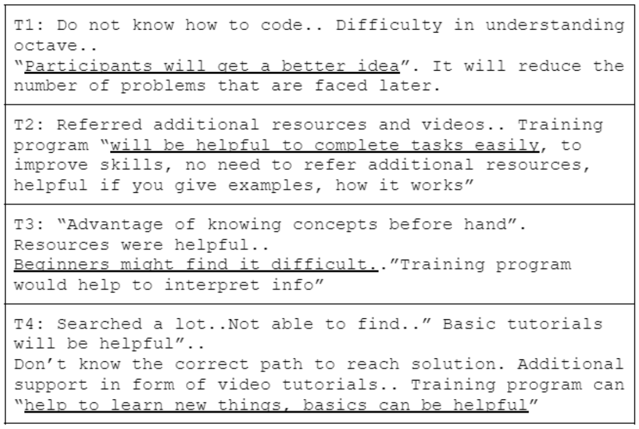
S3: "Referred to a lot of YouTube links and stackoverflow" that consumed time..No additional support is required

S4: It's a competition, no training is required.
Competitors will increase.
Self-learning should happen.."For learning, a training program is useful"

Figure 1. Screenshot of interview transcript done for individual interviews. S1 - Student 1, S2 - Student 2, S3 - Student 3, S4 - Student 4.

Following are few overall interview findings or insights:

- Low scorer teams lack knowledge so need training programs to learn basics.
- Top scorer teams either have learnt about domain knowledge through previous competition participation or done some courses so are able to submit the tasks.
- Training programs should contain theme-specific topic, coding.
- The training program should cover basics and should not clash with academics.
- The training program will be good for newcomers.



T1: Do not know how to code.. Difficulty in understanding octave..
"Participants will get a better idea". It will reduce the number of problems that are faced later.

T2: Referred additional resources and videos.. Training program "will be helpful to complete tasks easily, to improve skills, no need to refer additional resources, helpful if you give examples, how it works"

T3: "Advantage of knowing concepts before hand".
Resources were helpful..
Beginners might find it difficult.."Training program would help to interpret info"

T4: Searched a lot..Not able to find.." Basic tutorials will be helpful"..
Don't know the correct path to reach solution. Additional support in form of video tutorials.. Training program can "help to learn new things, basics can be helpful"

Figure 2. Screenshot of interview transcript done for team interviews. T1 - Team 1, T2 - Team 2, T3 - Team 3, T4 - Team 4.

B. Survey Form:

Apart from interviews, a survey form was designed to collect data from larger groups of participants. It was to understand the need, topics, and duration of the training program. Total 2012 responses were received. Following are few insights:

1) Students were asked if there is a need for a training program before the start of competition. Figure 3 shows that 1645 (81.75%) students out of 2012, expressed the need for a training program. This resembles the responses received through interview. Students esp. from low and

medium scorer teams expressed that they are completely new to the topics introduced in the competition. As they lack basics, most of their time is invested in going through the resources and learning and less time is left to work on the tasks which leads to missing out on deadlines.

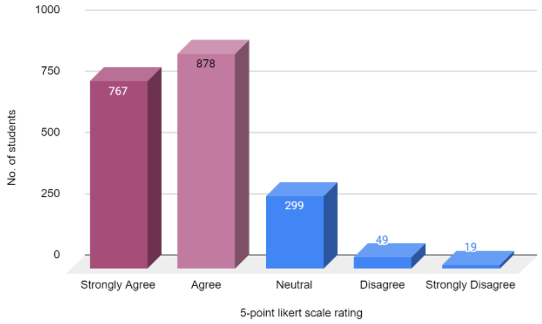


Figure 3. Need of training program

2) Students were also asked to rate if they referred to a lot of external resources while working on the tasks during competition.

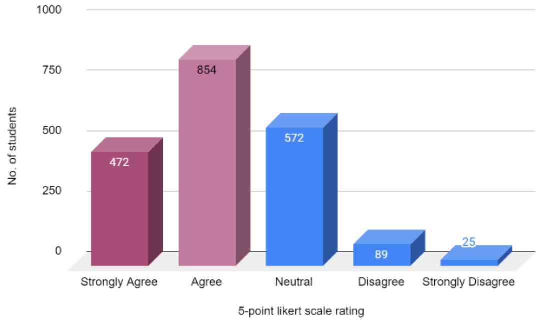


Figure 4. Referred external resources

Figure 4 shows that 1326 (65.90%) students felt that provided resources were inadequate and that made them refer to additional resources. This may be because as students lack basics they look out for more information.

3) Students were further asked if the training program at the start of competition would help them and how it would help them. Figure 5 shows the result for the former part where 1687 (83.84%) students responded that it would be helpful.

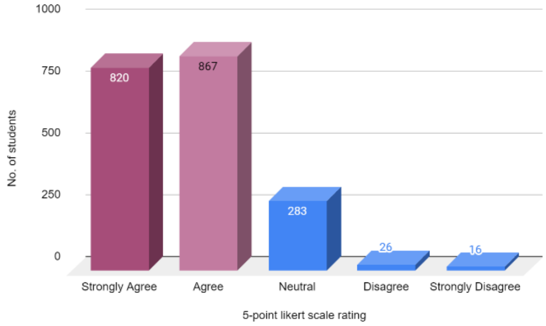


Figure 5. Help of training program

For the latter part of the question, text responses were analyzed. Out of 2012 responses, 512 were read thoroughly. Major responses stated the training program would help them to understand basics, gain knowledge, understand tasks better in competition, would be beneficial for newcomers, preparation before the competition would save time during competition which will help them to meet deadlines and not drop out and few responses were related to thinking and problem-solving skills. Above data collection and analysis helped me understand that students do need a training program and would be beneficial for reasons stated above.

To address Goal 2, further analysis of the survey responses was done. Following are the few insights:

1) Figure 6 shows the results for the duration of the training program. Student responded that the training program should either be less than or equal to 4 weeks. This may be because the competition is already 7-8 months long competition. Having a program more than 4 weeks will make it difficult for the students to manage their other activities. As per interview response from few teams, having the training program before the competition would be beneficial for them as they have summer break during that slot. So, a four-week program will not interfere with their academics.

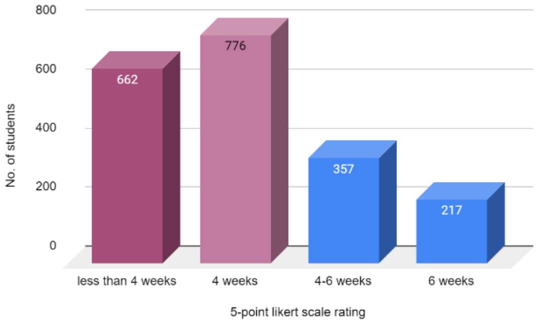


Figure 6. Duration of the training program

2) When asked if students would like to attempt a training program, the results obtained are as shown in Figure 7.

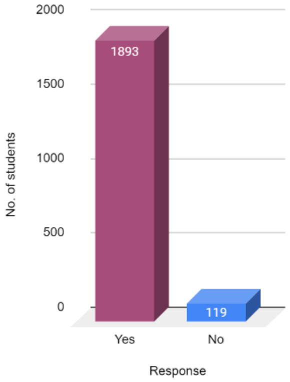


Figure 7. Attempt Training program

3) Students were also asked about what they think which topics should be covered in the training program. They responded with varied domains with most frequently

occurring Robotics and Embedded System (17.59%), Image processing (6.46%), Python (4.72%) and others were more theme specific topics. Student response is like the analysis done for domains covered in past years of e-Yantra Robotics Competition.

Above analysis leads to the conclusion that it is essential for students to get acquainted with different robotics concepts and research says that it should be taught through problem solving activities. According to ABET-mapped competencies (problem solving, communication, teamwork, ethics, life-long learning, math, science, engineering knowledge; engineering tools; experiments and data, design, contemporary issues, understand impacts), problem solving is an essential competence for undergraduates in engineering domain [6]. Authors have also identified problem solving as one of the important learning outcomes for Educational Robotics competition [2]. Various authors like Jonassen, Polya, Simon, Bransford and Stein, Hayes and Sternberg have proposed different problem-solving strategies that can be used while designing problem solving activities for the training program. Technical paper [7] states different principles that form the basis of problem solving in classroom or computer-based settings.

Work for Goal 3 is in the planning stage. We are planning to design an online training program that will be made available in the online mode and would include video tutorials, problem solving activities based around robotics and quiz.

3. ADVICE SOUGHT

Out of the above three stated goals, work for goal 1 is accomplished whereas for goal 2 is in progress. I need feedback on the work done towards two goals. For goal 3, I aim to design and implement the training program. I plan to collect the following data at three different instances through the training program:

1) Start: Pre-Questionnaire (this will give me an understanding of their prior knowledge)
2) During: Videos watched, problem solving activities, quiz attempted (this will give me feedback on the module wise content)
3) End: Semi-structured interviews, Post Questionnaire (feedback to their experience to further improvise the program)

This data will give feedback for the training program. The effectiveness of the program will be measured in competition with two groups (control and experimental). Research is at the early stage, and I hope the consortium can provide suggestions on following two questions:

1) What more data can be collected through the training program?
2) What are the analysis techniques that can be used?

4. ACKNOWLEDGMENTS

The authors thank the Ministry of Education (MoE) for funding the e-Yantra project. We gratefully acknowledge the support of e-Yantra project staff in devising the robotics competition and participants for interviews and feedback.

5. REFERENCES

[1] Angel-Fernandez, Julian & Vincze, Markus. (2018). Towards a Formal Definition of Educational Robotics. 37-42. 10.15203/3187-22-1-08.

[2] S. Evripidou, K. Georgiou, L. Doitsidis, A. A. Amanatiadis, Z. Zinonos and S. A. Chatzichristofis, "Educational Robotics: Platforms, Competitions and Expected Learning Outcomes," in IEEE Access, vol. 8, pp. 219534-219562, 2020, doi: 10.1109/ACCESS.2020.3042555.

[3] S. Krithivasan et al., "Learning by competing and competing by learning: Experience from the e-Yantra Robotics Competition," 2014 IEEE Frontiers in Education Conference (FIE) Proceedings, Madrid, Spain, 2014, pp. 1-8, doi: 10.1109/FIE.2014.7044136.

[4] “World Robot Olympiad (WRO) - India,” [Online]. Available: https://wroindia.org/ [Accessed on: January 20, 2023.]

[5] “ABU Robocon,” [Online]. Available: http://www.aburobocon2022.com/ [Accessed on: January 20, 2023.]

[6] Passow, H.J. and Passow, C.H. (2017), What Competencies Should Undergraduate Engineering Programs Emphasize? A Systematic Review. J. Eng. Educ., 106: 475-526. https://doi.org/10.1002/jee.20171.

[7] Foshay, Wellesley & Kirkley, Jamie. (1998). Principles for Teaching Problem Solving.

[8] R. Ekatpure, S. Jadhav, K. Karia and K. Arya, "Musical Mimicry to Learn Audio Processing," 2019 IEEE Tenth International Conference on Technology for Education (T4E), Goa, India, 2019, pp. 214-217, doi: 10.1109/T4E.2019.00048.

[9] K. Karia, R. Bessariya, K. Lala and K. Arya, "Learning While Competing - 3D Modeling & Design," 2018 IEEE Tenth International Conference on Technology for Education (T4E), Chennai, India, 2018, pp. 93-96, doi: 10.1109/T4E.2018.00026.

Investigating teams' Socially Shared Metacognitive Regulation (SSMR) and transactivity in project-based computer supported collaborative learning environment

Vishwas Badhe
Indian Institute of technology Bombay
vishwasbadhe@iitb.ac.in

Chandan Dasgupta
Indian Institute of Technology Bombay
cdasgupta@iitb.ac.in

Ramkumar Rajendran
Indian Institute of Technology Bombay
ramkumar.rajendran@iitb.ac.in

ABSTRACT

In collaborative project-based learning environments, students handle ill-structured challenges and practice socially shared metacognitive regulation (SSMR). Transactivity refers to the degree to which students demonstrate a shared engagement and build on each other's knowledge contributions. Prior research has highlighted the need to investigate SSMR and transactivity systematically. Putting learners in a team and assigning project does not guarantee the success of the collaboration. Collaborating team members may face cognitive and metacognitive issues due to different levels of metacognitive capabilities. To support SSMR and to have teams with a high level of transactivity, we need to understand the shared regulation behavior of team members. Interestingly, the lack of studies in this domain directed us to understand the shared regulation behavior of team members and their transactive interactions. We have conducted two studies which are primarily focusing on qualitative data. To validate and triangulate the claims using another mode of data, we are proposing an additional mode of data i.e. system interaction data. Based on our understanding, further in our research goals, we are proposing a computer-supported learning environment to foster SSMR and a higher level of transactivity. We will try to achieve this through metacognitive prompts as scaffolds for team members. We present the initial work done in this direction and we proposed one additional mode of data. Currently, most of the learning environments are focusing on individual learners, so we are trying to bridge this gap through the proposed system, supporting SSMR & transactivity in a project-based CSCL context. We intend to seek advice on the validity and reliability of our approach to understand SSMR & transactivity and further measure its impact on collaborating teams.

Keywords

Socially shared regulation of learning (SSRL), Socially shared metacognitive regulation (SSMR), Transactivity, CSCL, Project-based learning, open-ended problem, collaborative problem solving (CPS).

1. INTRODUCTION

Computer-supported collaborative learning environments (CSCL) facilitate interactions among learners to acquire knowledge, skills, and attitudes [2, 7, 8]. As learners are coming from diverse socio-cultural backgrounds, they bring diverse goals, approaches, attitudes, and experiences which become an important and dynamic element in collaborative learning environments. Handling the dynamic nature of the team and simultaneously achieving progress in a given task needs many socially shared regulation strategies amongst the collaborating members [6]. While collaborative learning looks attractive for facilitating collective knowledge construction, it's not easy to orchestrate [10]. While working collaboratively on the set of tasks, some cognitive and metacognitive issues may arise due to differences in task and content understanding or different interpretations of the task by different learners [5].

To ensure the success of collaboration, learners must develop a shared mental model and a collective scheme of cognitive interdependence for communication and coordination to derive the high-quality participation of each team member in the shared task [6]. Metacognition plays a vital role in collaboration to make members aware of the challenges and need for regulation. Socially shared metacognitive regulation (SSMR) is an important process in collaborative learning which refers to participants' goal-directed, consensual, egalitarian, and complementary regulation of joint cognitive processes in the collaborative learning context [3, 4]. SSMR ensures the appropriate direction of the groups' cognitive activity using constant monitoring and controlling of the cognitive process.

A recent study illuminates that SSMR has some relation with idea of transactivity which refers to reciprocity and interdependence in the transactions between learning partners and between those partners and the task [1].Transactive discussion refers to a type of verbal interaction in which each learner uses own conversational turn to operate on the reasoning of the partner or to clarify his or her own ideas [12]. The scale of transactivity comprises different social modes of co-construction and represents different degrees of transactivity. On this scale, externalization and quick consensus building is regarded as the least transactive social mode, whereas conflict-oriented consensus building is the most transactive social mode [12].

In the interconnected and interdisciplinary knowledge-driven professional environment, the ability to work collaboratively on ill-structured long-term project goals (e.g Global Goals - https://www.globalgoals.org/) and engaging in socially shared regulated learning throughout the process have become vital skills. In this context, we explore project-based learning for fostering such socially shared regulation of learning (SSRL). Project-based learning pedagogy has six features - (a) learning goals, (b) collaboration, (c) focus question, (d) engagement in scientific practice, (e) scaffolding with learning technology, and (f) creation

V. Badhe, C. Dasgupta, and R. Rajendran. Investigating teams' socially shared metacognitive regulation (ssmr) and transactivity in project-based computer supported collaborative learning environment. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 552–555, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115657

of tangible solutions useful for addressing real-world problems [9]. In project-based learning, learners engage with the problem, learn by doing, discussing, applying ideas and try to solve the problem given to them, which increases learners' engagement and helps them to develop a deeper understanding of important ideas by facilitating them opportunities for problem-solving, decision-making, and explaining their ideas [9].

To ensure success of project-based learning in the CSCL environment, we are focusing on with socially shared regulation for team members while working collaboratively. Team members use cognitive and metacognitive strategies while working, so we have investigated their SSRL & SSMR strategy application and level of transactivity in two studies. Understanding the shared regulation processes is important in order to support their regulation in the context of project-based learning in the CSCL environment. Following are our research goals (RGs)

•RG1: Conduct studies to understand learners' socially shared regulation behavior in project-based CSCL environments.

•RG2: Design and develop a learning environment to foster socially shared metacognitive regulation (SSMR) using metacognitive prompts in a project-based learning context.

•RG3: Measure and validate the impact of metacognitive prompts given in the learning environment to foster SSMR and transactivity in a project-based CSCL

2. RESEARCH PROGRESS

The research progress till now is given in this section. We have explained each research progress with respect to each research goal as follows:

RG 1: Conduct studies to understand learners' socially shared regulation behavior in project-based CSCL environments.

In order to address this research goal, we conducted two research studies as detailed below.

Study 1: The objective of this study was to understand the Socially shared regulation of learning (SSRL) strategy application by teams. In the first study, the differences in application of SSRL strategies were studied for high and low performing teams in project-based learning settings having open-ended problems (tasks). We have found considerable differences in the application of SSRL strategies between high and low-scoring teams, those differences were represented by using quantitative and thematic representations. For analyzing the data, we have used the framework given by [10].

Study 2: The first objective of this study was to understand the socially shared metacognitive regulation (SSMR) which is a subcomponent of SSRL and one type of regulation learners use in SSRL context. The second objective of this study was to understand the relation between SSMR & transactivity. We found a considerable difference in application of SSMR strategies by teams, which highlight some important aspects of the relationship between SSMR and transactivity. Findings about this dynamic relationship are reported through quantitative and thematic representations. To analyze the SSMR strategy, we have used the framework given by [4] and for analyzing the transactivity externalized through verbalized interactions, we have used the frame-work given by [12]. Two Studies were conducted as part of 12 weeks of a graduate-level face-to-face semester-long course having an open-ended problem statement. Participants were divided into teams consisting of 4 members each; each team consisted of Master's, Ph.D., and Bachelor's level learners. The course followed a project-based learning approach and was divided into major milestones leading to the final solution. For each week, learners were given one hour for teaching by an instructor and two hrs for teamwork. At the start of each milestone, each team collectively responded to the OurPlanner tool and at the end of each milestone, each team collectively responded to the OurEvaluator tool. At the end of each milestone, teams were asked to present their team progress to the entire class. Teams were instructed to log their progress in shared group journals asynchronously. Group interactions were video recorded. To investigate the SSMR and degree of transactivity in those contrasting teams, we analyzed the video data (15 hours) from a synchronous face-to-face classroom. The content analysis approach was followed to analyze students' verbalized interactions in high and low-performing teams to see emergent relationship between SSMR and transactivity. The team's performance was evaluated by a predefined rubric. The video was segmented into episodes that map to multiple conversational turns by multiple students while they were working on various topics. Those episodes were considered SSMR episodes if verbalizations were referred to as monitoring and controlling cognitive processes [1].

In both studies, we tried to investigate the SSRL & SSMR strategy applications of learners from high and low performing teams. Along with that we have investigated the relationship between SSMR & transactivity of teams while working in project-based learning settings having open-ended problems (tasks). In proposed research, three major parameters of SSMR are considered to quantify the SSMR episodes. a) Metacognitive regulation skill used in SSMR episode (orienting, planning, monitoring, reflection), b) Focus of SSMR episode (Fundamental, organizational, surface level), c) function of SSMR episode (Facilitate or inhibit the current metacognitive activity). The data from both the studies were mostly qualitative in nature and were analyzed by manual method (ground root) using established frameworks. We have reported the differences between teams using Quantitative and thematic representations.

So far the modes of data we have collected were a) video & audio data of teams while working collaboratively b) Self-reported data by team members, and c) performance of teams. The evidence we have collected to support the claims were based on these data sources. As per the existing literature, most of the studies have investigated SSMR for mathematics domain, so they have used mathematics word problem specific parameters while investigating SSMR. Some studies have collected gesture and GSR data to investigate, but these methods are mostly used in small duration studies. As proposed study was face-to-face and longitudinal in nature hence it was not feasible to use these data modes because learners were supposed to move physically and interact with other participants. The proposed study design intends to investigate SSMR for collaborative programming tasks (using open-ended project based learning pedagogy) using verbal interaction data and some extent of self-report data such as surveys and interviews. Hence as of now verbal interaction data and self-report data are two most feasible modes available for investigating SSMR for collaborative programming tasks.

The existing study design followed for above two studies is represented in the fig 1 which shows different data modes. For the teams working in project-based learning settings and having open-ended problems (tasks), we have derived understanding about the teams' SSMR strategy application and level of transactivity teams

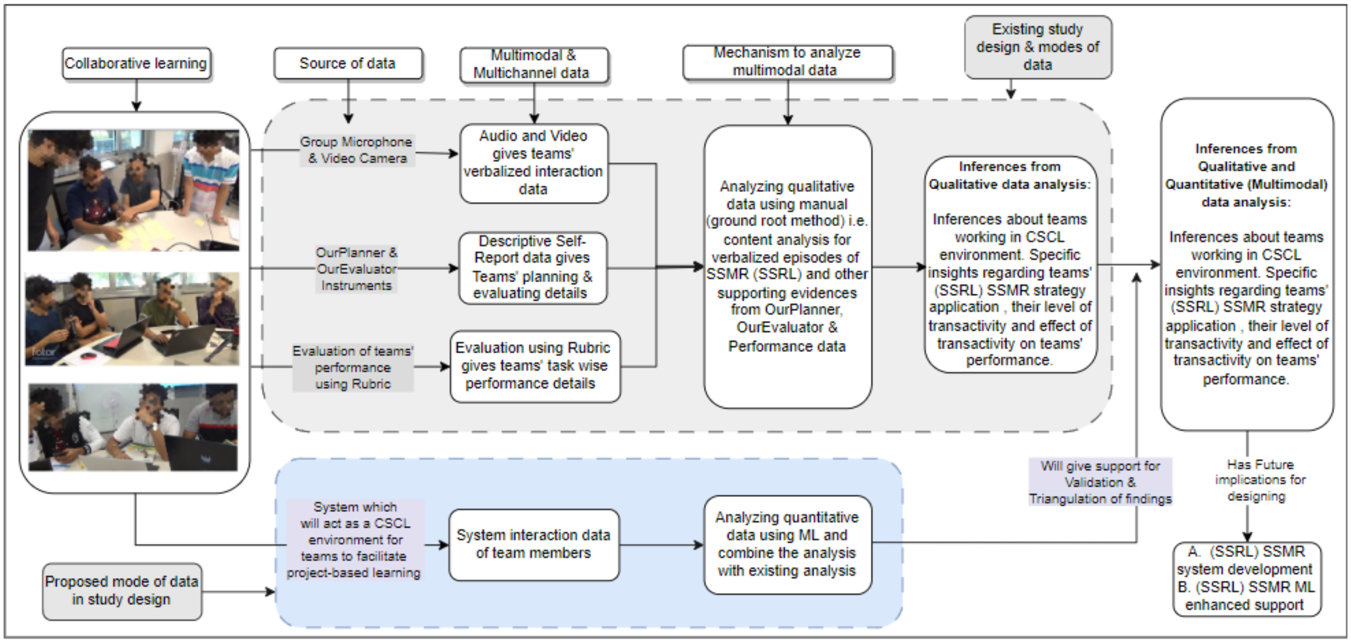

Collaborative learning
Source of data
Multimodal & Multichannel data
Mechanism to analyze multimodal data
Existing study design & modes of data
Group Microphone & Video Camera
Audio and Video gives teams' verbalized interaction data
OurPlanner & OurEvaluator Instruments
Descriptive Self-Report data gives Teams' planning & evaluating details
Evaluation of teams' performance using Rubric
Evaluation using Rubric gives teams' task wise performance details
Analyzing qualitative data using manual (ground root method) i.e. content analysis for verbalized episodes of SSMR (SSRL) and other supporting evidences from OurPlanner, OurEvaluator & Performance data
Inferences from Qualitative data analysis:
Inferences about teams working in CSCL environment. Specific insights regarding teams' (SSRL) SSMR strategy application , their level of transactivity and effect of transactivity on teams' performance.
Inferences from Qualitative and Quantitative (Multimodal) data analysis:
Inferences about teams working in CSCL environment. Specific insights regarding teams' (SSRL) SSMR strategy application , their level of transactivity and effect of transactivity on teams' performance.
Will give support for Validation & Triangulation of findings
Has Future implications for designing
System which will act as a CSCL environment for teams to facilitate project-based learning
System interaction data of team members
Analyzing quantitative data using ML and combine the analysis with existing analysis
Proposed mode of data in study design
A. (SSRL) SSMR system development
B. (SSRL) SSMR ML enhanced support

Figure 1. Existing study design with proposed mode of data

attain. The proposed project is intended to model the SSMR behavior of team members while working on open-ended collaborative programming projects. The SSMR model will inform collaborative system developers for programming tasks and teachers who use project based learning pedagogy in programming projects in course. On the basis of the understanding, we are proposing a framework of learning environment.

RG2: Design and develop a learning environment to foster socially shared metacognitive regulation (SSMR) using metacognitive prompts in a project-based learning context.

To provide a prompt as a scaffold for a learner (based on what we learned from above 2 studies), we searched for existing learning environments which provide prompts as scaffold for learners in project-based CSCL context. Since there is not much extensive research available on such learning environments to foster learners' SSMR, we propose to design and develop a learning environment. The proposed learning environments will also try to overcome the limitation of not having multiple modes of data. Currently the data source is mainly video and audio collected through manual way.

In order to validate our findings from research studies we propose to collect data from the learning environment and triangulate the findings. The data collected from the learning environments will be used to support our claims regarding teams while they are applying SSMR strategies. In our learning environment, we will record the learners' interaction data generated from click-stream. So while team members will be working collaboratively on open-ended problems in project based learning, they will be contributing to the shared goal from their own system. The learning environment will help us in collecting multimodal data in addition with the video and audio data. We plan to develop a system that

captures the learners' interaction behavior. The learners actions along with timestamp will be analyzed to understand the SSMR. We propose to use Process mining tools like ProM to analyze the data. In order to support the SSMR in collaborative learning environment, we are willing to use ML algorithms to detect the place in SSMR to provide scaffold. We are trying to achieve triangulation while establishing our claims through multiple modes of data. The proposed mode of data is also highlighted in fig 1.

RG3: Measure and validate the impact of metacognitive prompts given in the learning environment to foster SSMR and transactivity in a project-based CSCL context.

This is the final research goal after conducting studies with a learning environment. The major focus here will be to measure and validate the impact of metacognitive prompts given in the learning environment to foster SSMR and transactivity in a project-based CSCL context.

3. ADVICE SOUGHT

Question 1: Is the study design with proposed mode of data capable/suitable for validation/triangulation of research claims?

Question 2: Is the proposed mode of data (system interaction) aligned with existing modes of data? If not what are the ways to make it aligned for given research goals?

Question 3: How to handle overlapping areas of two different modes of data (i.e. audio-video and system interaction)?

4. CONCLUSION

On the basis of understanding about SSRL, SSMR and transactivity of teams while working in project-based learning in CSCL context, we intend to add one more mode of data channel (i.e. System interaction data from proposed learning environment). This will help us to validate and triangulate our claims with evidence from multiple modes. In order to make collaborative project-based learning successful, we need to understand the (SSRL) SSMR process and teams' transactivity in detail using data from multiple channels. Here, we are proposing that we need a learning environment to collect multi modal data to understand teams' regulation behavior and ultimately to support collaborating teams with metacognitive prompts. As there is not much intensive research that has happened on supporting teams' regulation while working in project-based CSCL environments, our proposed learning environment may help teams to regulate better and have a high level of transactivity. Because we have understood from our studies, applying maximum (SSRL) SSMR strategies and attaining a high degree of transactivity have high correlation with high performance.

Though we have some understanding from previous studies, it's based on some assumptions like, a) team members may have externalized their potential metacognitive strategy application capability while working collaboratively, b) team members may have worked on problem statements in project-based learning in class, when data was collected etc. Considering the assumptions, those are limitations for this research. We feel that this research process is at the defining moment of its journey and seeking some advice for the future discourse with respect to some challenges. We request feedback from experts in this community to overcome/handle challenges so that the proposed learning environment can be developed and impact the teams' (SSRL) SSMR and transactivity in an effective way.

5. REFERENCES

[1] De Backer, L., Van Keer, H., & Valcke, M. (2022). The functions of shared metacognitive regulation and their differential relation with collaborative learners' understanding of the learning content. Learning and Instruction, 77, 101527.

[2] Dillenbourg, P. (Ed.) (1999). Collaborative learning: Cognitive and computational approaches. Oxford, U.K.:Pergamon.

[3] Iiskala, T., Vauras, M., Lehtinen, E., & Salonen, P. (2011). Socially shared metacognition of dyads of pupils in collaborative mathematical problem-solving processes. Learning and instruction, 21(3), 379-393.

[4] Iiskala, T., Volet, S., Lehtinen, E., & Vauras, M. (2015). Socially shared metacognitive regulation in asynchronous CSCL in science: Functions, evolution and participation. Frontline Learning Research, 3(1), 78-111.

[5] Järvelä, S., & Jarvenoja, H. (2011). Socially constructed self-regulated learning and motivation regulation in collaborative learning groups. Teachers College Record, 113(2), 350–374.

[6] Järvelä, S., Hadwin, A., Malmberg, J., & Miller, M. (2018). Contemporary perspectives of regulated learning in collaboration. In International handbook of the learning sciences (pp. 127-136). Routledge.

[7] Kaye, A. (1995). Computer-supported collaborative learning in a multi-media distance education environment. In C. O'Malley (Ed.), Computer supported collaborative learning (pp. 125–144). Berlin: Springer-Verlag.

[8] Koschmann, T, (Ed.)(1996). CSCL: Theory and practice of an emerging paradigm. Mahwah, NJ: Lawrence Erlbaum Associates.

[9] Krajcik, J. S., & Shin, N. (2014). Project-based learning. In R. K. Sawyer (Ed.), The Cambridge handbook of the learning sciences (2nd ed.) (pp. 275–297). New York, NY: Cam-Cambridge University Press.

[10] Lobczowski, N. G., Lyons, K., Greene, J. A., & McLaughlin, J. E. (2021). Socially shared metacognition in a project-based learning environment: A comparative case study. Learning, Culture and Social Interaction, 30, 100543.

[11] Panadero, E., & Järvelä, S. (2015). Socially shared regulation of learning: A review. European Psychologist. doi: 10.1027/1016-9040/a000226

[12] Teasley, S. (1997). Talking about reasoning: How important is the peer in peer collaboration? In L. B. Resnick, R. Säljö, C. Pontecorvo, & B. Burge (Eds.), Discourse, tools and reasoning: Essays on situated cognition (pp. 361–38)

Fostering Interaction in Computer-Supported Collaborative Learning Environment

Pratiksha Virendra Patil
Indian Institute of Technology Bombay
214380005@iitb.ac.in

Ashwin T S
Indian Institute of Technology Bombay
ashwindixit9@gmail.com

Ramkumar Rajendran
Indian Institute of Technology Bombay
ramkumar.rajendran@iitb.ac.in

ABSTRACT

Interaction is the key driving force behind the critical processes involved in collaborative learning. But novice learners find it difficult to effectively interact during collaborative tasks and need support. Speech data from the collaborative discourse is significantly used to monitor and assess the interaction among students. Our research goal is to foster interaction in computer supported collaborative learning environment. The initial plan is to design and develop a system for capturing speech data from collaborative discourse in real-time, and derive various verbal and non-verbal features from speech data to automatically detect and assess the collaboration quality. Also, we are planning to design and implement the real-time automated feedback based on the data captured and investigate the impact of the feedback provided.

Keywords

Collaborative Learning, Automatic detection of collaboration, Speech

1. INTRODUCTION

Collaborative learning is one of the 4C skills, considered to be the most important 21st-century learning skill. Much emphasis is being provided on enhancing collaborative learning skills in students and the workforce [1, 2]. The construction of shared knowledge, negotiation/coordination, and maintaining team function are critical processes involved in collaborative learning. Interaction among students is the driving force behind these meaning-making processes [3, 4, 5]. Collaboration is key to learning but it is not easy for novice learners to effectively interact in a collaborative task. Lack of interaction is due to challenges like students not being open in accepting opposing views, asking for help, building trust and giving elaborate explanations or providing feedback [6]. Cognition, affect, motivation, and metacognition of individual participants and group members also plays a role in interaction [7]. Interaction in collaborative learning is also influenced by factors such as group dynamics, pedagogy, task design, and process of evaluation as well [8]. To address the lack of interaction, appropriate support needs to be provided to the students. And such support can be provided only after a timely and accurate diagnosis of the challenges faced by students [4, 8, 9, 10, 11].

Generally, the quality of collaboration is measured across five aspects as follows,1) communication/appropriate use of social skills; 2) joint information processing/group processing; 3) coordination/positive interdependence; 4) interpersonal relationship/promotive interaction; and 5) motivation/individual accountability. These five aspects are further mapped to the nine dimensions of collaboration, those are expertise, dominance, coupling, reflection, roles, engagement, coherence, misconception and uncertainty. These dimensions are measured largely using self-reports or conducting tests. But it has its own limitations. Quality assessment based on observation can be leveraged to address these limitations. Monitoring students' discourse can give clarity on student understanding and challenges faced by them while working on collaborative tasks [12]. To monitor the collaborative discourse, the existing researchers are largely dependent on manual observation, transcription, and analysis to identify challenges faced by learners. This is a time-consuming and laborious process that also causes delays in feedback. Moreover, it puts limitations on scaling collaborative learning activities [13, 14].

Recently there are a lot of research studies focused on providing adaptive support in computer-mediated/online collaborative activities by monitoring discourse from forum posts, chats, and log data [15]. Some researchers also have used multimodal data from online meetings to assess collaboration and provide feedback [16]. However, research in automated monitoring of collaborative discourse in physical spaces (collocated collaboration) to assess collaboration dynamically in real-time and provide adaptive feedback [17] is still in a nascent stage and emerging rapidly with the advancement in sensor technology and in the field of multimodal learning analytics. Multimodal data like gestures, posture, eye gaze, content, log data, self-reports, spatial data, facial expressions, and physiological indicators [18] can help us measure important collaboration indexes such as synchrony in

P. V. Patil, A. T. S, and R. Rajendran. Fostering interaction in computer-supported collaborative learning environment. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 556–560, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115798

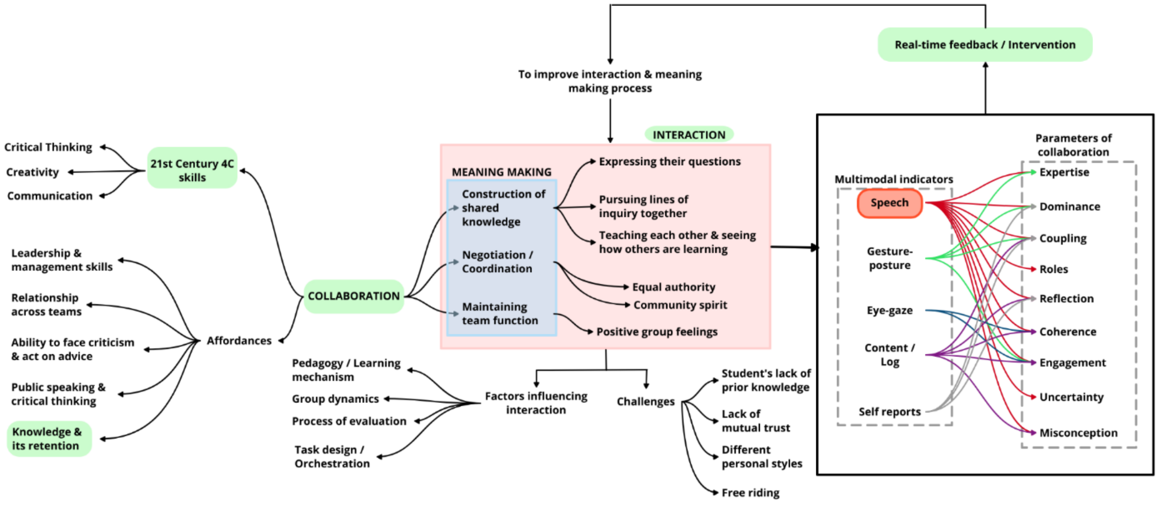

Real-time feedback / Intervention
To improve interaction & meaning making process
INTERACTION
Critical Thinking
Creativity
Communication
21st Century 4C skills
MEANING MAKING
Construction of shared knowledge
Negotiation / Coordination
Maintaining team function
Expressing their questions
Pursuing lines of inquiry together
Teaching each other & seeing how others are learning
Equal authority
Community spirit
Positive group feelings
Leadership & management skills
Relationship across teams
Ability to face criticism & act on advice
Public speaking & critical thinking
Knowledge & its retention
Affordances
COLLABORATION
Pedagogy / Learning mechanism
Group dynamics
Process of evaluation
Task design / Orchestration
Factors influencing interaction
Challenges
Student's lack of prior knowledge
Lack of mutual trust
Different personal styles
Free riding
Multimodal indicators
Speech
Gesture-posture
Eye-gaze
Content / Log
Self reports
Parameters of collaboration
Expertise
Dominance
Coupling
Roles
Reflection
Coherence
Engagement
Uncertainty
Misconception

Figure 1. Conceptual representation of the proposed work.

members of the group, equality of contribution, information pooling, and so on. For example, the rise and fall of average pitch, intensity obtained from the audio signal; body position, head direction, pointing, using both hands, joint visual attention, etc.; these indicators can be used to measure synchrony. High synchrony indicates good collaboration. Total speaking time and the number of ideas and questions raised can tell us about equality in participation, which is indicator of good collaboration. Web search is the indicator for information pooling, similarly, different multimodal indicators are mapped to different collaboration indexes. These indexes can further help us understand high-level collaboration parameters like expertise, dominance, coupling, coherence, roles, reflection, uncertainty, misconception, and engagement. It is also evident that all these aspects of collaboration can be significantly detected using speech modality [18]. Mapping of audio indicators with different parameters of collaboration is shown in Table 1.

Table 1. Mapping of audio indicators to collaboration parameters [18].

Collaboration parameters	**Collaboration Indexes**	**Audio Indicators**
Expertise, Dominance,Coupling,Reflection,Roles,Engagement,Coherence,Misconception,Uncertainty	Synchrony	rise and fall of average pitch,intensity, amplitude
	Equality	jitter,total speaking time
	Mutual Understanding	dialouge management,verbal discourse,statements,questions

In the speech, the existing works used verbal and non-verbal features to understand various aspects of collaboration. It is clear that cognitive and socio-emotional interaction can be analysed significantly using speech. There are works related to understanding the semantics of the interactions and the automation of the same is not explored much in the literature. Moreover, capturing multimodal data in a live classroom to assess the quality of collaboration dynamically in real-time and providing adaptive feedback is a challenging task. Such feedback can help students collaborate better in face-to-face settings [19]. Also, it will help in reducing the cognitive load on teachers/facilitators and enable them to effectively conduct collaborative classes on a large scale [16, 19, 20]. From the existing studies, we observe that very few works attempted automatic detection of collaboration using multimodal data and specifically using speech alone. And those are limited to very few aspects of collaboration. There are a lot of scopes to leverage non-verbal and verbal features of speech in the detection and assessment of collaboration. Non-verbal features like duration of speech, pause, turn-taking, pitch, jitter, intensity etc. can be used to detect low level collaboration indexes like equality and synchrony.

Non-verbal features extracted from speech can help in assessing collaboration quality in diverse contexts and tasks, while preserving the privacy of participants [28, 29]. On the other hand verbal/lexical features extracted from speech data captured during student collaborative discourse can be effectively used to create knowledge graphs and analyse them deeply to understand conceptual knowledge and transactivity between concepts [22, 23]. This understanding can lead to the design of effective feedback to foster cognitive interaction in the collaborative learning task. We are planning to address a few of these gaps in our proposed work. The conceptual framework of the proposed work is shown in figure 1. We are planning to design a system for capturing speech data from collaborative discourse in real-time, and derive various verbal and non-verbal features from speech data to automatically detect and assess the collaboration quality. Also to design and implement the feedback based on the data captured and investigate the impact of the feedback provided.

2. METHODOLOGY AND PROGRESS

In order to understand how students collaborate and also to learn the processes involved in our proposed framework, we conducted a study to capture speech data from a collaborative learning environment. We used data from 12 participants solving a programming problem with a shared screen using a python programming teaching environment. Students worked in dyads.

There were six groups in total. Before the study test was conducted to check the knowledge of participants of basic Python Programming. It had 12 multiple choice questions and 2 questions, for which they were required to write complete code. This study was conducted in a technology-enhanced collaborative learning classroom. (Refer to section 5.2)

2.1 Data Collection and Automated Speech Transcription

Introduction to the learning environment and all task-related instructions were provided by the instructor. Students had access to the study material while solving the task. Students' video and speech and log data were captured using the Open Broadcaster Software (OBS) installed on the computer they were using. Students' speech data captured using the OBS tool, is then converted into transcripts using a web-based tool for transcription, i.e., Otter.ai. Each group's audio files were automatically transcribed using the web-based Speech to text service. The service generates a transcript with a timestamp for every utterance spoken along with speaker diarization. It also provides a summary of keywords and the total speaking time of each speaker in percentage. It is observed that automated transcription has some limitations like, not capturing overlap between two speakers, jumbling with similar pronouncing words, and not fully transcribing long sentences. However, as we are interested in knowing who the speaker is and what topics are discussed, these errors will not have a significant impact [5]. We generated a transcript for an audio file containing spoken interaction between dyads. It was a 50 minutes collaborative problem-solving (Python Programming) activity. Further, we have coded the transcripts by marking co-occurrences of keywords in each utterance. We divided transcripts based on its timestamp into 5 scenes and compared the data for each speaker. In our study, we focused on the semantics/content of verbal data. We looked at the contribution of each group member in terms of keywords/concepts discussed by them, Speaker identification and timestamp in order to map turn-taking and overlap of speech. The Epistemic Network model for this data is created using web-based tool [21, 22, 23, 24]. Creating knowledge graphs to understand unfolding collaboration automatically in real time is a challenging task. Speech data needs to be converted into text/transcripts using NLP to understand the semantics of data. We will be using speaker diarization and text data mining to understand who is speaking, when they are speaking and most importantly what they are speaking- the concepts and linkages between the concepts. We are exploring how epistemic network analysis can be effectively used for this task and the need to incorporate social network analysis.

2.2 Audio signal processing for feature extraction

In order to detect collaboration indexes and asses collaboration quality based on speech interaction, we have extracted acoustic features like fundamental frequency, Mel-frequency cepstral coefficients (MFCCs) and pitch from audio file (wav file). This audio file contain 45 minutes of speech data captured during collaborative design task, in which 4 students are working on concept map and facilitator is providing the instructions. Audio signal is pre-processed by sampling it at the rate of 44 KHZ [30]. Then it is segmented for fixed time window and features are extracted using librosa, package in python for audio analysis. We have also clustered the data for different speakers using extracted features. Our aim is to use automated speaker recognition in multiparty audio files, segment the audio file for each turn of the speaker, extract acoustic and verbal features from the segments, and use these features to detect collaboration indexes.

3. CONTRIBUTION AND FUTURE WORK

We explored several studies that collect multimodal data from collaborative learning environments, we observe that most existing works analyse the quality of collaboration and some of them also provide feedback based on the assessment. After analysing the current literature we have identified certain gaps. Very few studies provide real-time dynamic feedback to the learners or facilitator, especially in the collocated collaborative learning environment. Most studies using speech modality to assess collaboration quality and provide feedback consider acoustic and non-lexical/ non-verbal features to classify/ detect collaboration using machine learning. This can lead to concerns about the reliability of the results. There exist no studies that provide real-time feedback based on learners' verbal cues and their interaction log data. Most of the studies focus on detecting and supporting dominance or coherence. There is no existing work to automatically detect other collaboration indexes such as uncertainty, misconceptions, etc. which can be better mapped using verbal features of speech. Current studies are providing feedback to foster socio-emotional interaction in collaborative learning. To address the research gaps, we aim to develop a system to automatically assess the collaboration quality in real-time. This system will capture speech data from collaborative discourse in real-time, and derive various verbal and non-verbal features from speech data using state-of-the-arts methods. Further it will segment and annotate the stream of data automatically to map it with different collaboration assessment indexes. Also we aim design and implement the feedback based on the real-time assessment and investigate the impact of the feedback provided.

4. REFERENCES

[1] Chiruguru, Dr & Chiruguru, Suresh. (2020). The Essential Skills of 21st Century Classroom (4Cs). 10.13140/RG.2.2.36190.59201.

[2] 2015. PISA 2015 Collaborative Problem Solving Framework

[3] Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.

[4] Gerry Stahl, Timothy Koschmann, Dan Suthers (2010). Computer-supported collaborative learning: An historical perspective. http://gerrystahl.net/cscl/CSCL_English.pdf

[5] Stewart, A. E. B., Vrzakova, H., Sun, C., Yonehiro, J., Stone, C. A., Duran, N. D., Shute, V., & D'Mello, S. K. (2019). I Say, You Say, We Say. Proceedings of the ACM on Human-Computer Interaction, 3(CSCW), 1–19. https://doi.org/10.1145/3359296

[6] Ha Le, Jeroen Janssen & Theo Wubbels (2018) Collaborative learning practices: teacher and student perceived obstacles to effective student collaboration, Cambridge Journal

of Education, 48:1, 103-122, DOI: 10.1080/0305764X.2016.1259389

[7] Taub, M., Azevedo, R., Rajendran, R., Cloude, E. B., Biswas, G., & Price, M. J. (2021). How are students' emotions related to the accuracy of cognitive and metacognitive processes during learning with an intelligent tutoring system?. *Learning and Instruction*, *72*, 101200.

[8] Dillenbourg P. (1999) What do yuo mean by collaborative leraning?. In P. Dillenbourg (Ed) Collaborative-learning: Cognitive and Computational Approaches. (pp.1-19). Oxford: Elsevier

[9] Kirschner, P.A., Sweller, J., Kirschner, F. *et al.* From Cognitive Load Theory to Collaborative Cognitive Load Theory. *Intern. J. Comput.-Support. Collab. Learn* 13, 213–233 (2018). https://doi.org/10.1007/s11412-018-9277-y

[10] Lin, Feng & Puntambekar, Sadhana. (2019). Designing Epistemic Scaffolding in CSCL.

[11] Yingbo Ma, Mehmet Celepkolu, Kristy Elizabeth Boyer. Detecting Impasse During Collaborative Problem Solving with Multimodal Learning Analytics. Proceedings of the 12th International Learning Analytics and Knowledge Conference (LAK22), 2022, pp. 45-55.

[12] Bressler, D. M., Bodzin, A. M., Eagan, B., & Tabatabai, S. (2019). Using Epistemic Network Analysis to Examine Discourse and Scientific Practice During a Collaborative Game. Journal of Science Education and Technology, 28(5), 553–566. https://doi.org/10.1007/s10956-019-09786-8

[13] Emma L. Starr, Joseph M. Reilly, and Bertrand Schneider ems,@mail.harvard.edu, josephreilly@g.harvard.edu, bertrand_schneider@gse.harvard.edu (2018). Toward Using Multi-Modal Learning Analytics to Support and Measure Collaboration in Co-Located Dyads. https://repository.isls.org/bitstream/1/888/1/55.pdf

[14] Nöel, René & Riquelme, Fabián & Lean, Roberto & Merino, Erick & Cechinel, Cristian & Barcelos, Thiago & Villarroel, Rodolfo & Munoz, Roberto. (2018). Exploring Collaborative Writing of User Stories With Multimodal Learning Analytics: A Case Study on a Software Engineering Course. IEEE Access. PP. 1-1. 10.1109/ACCESS.2018.2876801.

[15] Chua, Y. H. V., Rajalingam, P., Tan, S. C., & Dauwels, J. (2019). EduBrowser: A Multimodal Automated Monitoring System for Co-located Collaborative Learning. Learning Technology for Education Challenges, 125–138. https://doi.org/10.1007/978-3-030-20798-4_12

[16] Cornide-Reyes, H., Riquelme, F., Monsalves, D., Noel, R., Cechinel, C., Villarroel, R., Ponce, F., & Munoz, R. (2020). A Multimodal Real-Time Feedback Platform Based on Spoken Interactions for Remote Active Learning Support. Sensors, 20(21), 6337. https://doi.org/10.3390/s20216337

[17] Vogel, F., Kollar, I., Fischer, F. *et al.* Adaptable scaffolding of mathematical argumentation skills: The role of self-regulation when scaffolded with CSCL scripts and heuristic worked examples. *Intern. J. Comput.-Support. Collab. Learn* **17**, 39–64 (2022). https://doi.org/10.1007/s11412-022-09363-z

[18] Sambit Praharaj, Maren Scheffel, Hendrik Drachsler, Marcus Specht ; Literature Review on Co-Located Collaboration Modeling Using Multimodal Learning Analytics—Can We Go the Whole Nine Yards?.,IEEE Transactions on Learning Technologies,2021

[19] Lämsä, J., Hämäläinen, R., Koskinen, P., Viiri, J., & Mannonen, J. (2020). The potential of temporal analysis: Combining log data and lag sequential analysis to investigate temporal differences between scaffolded and non-scaffolded group inquiry-based learning processes. Computers & Education, 143, 103674. https://doi.org/10.1016/j.compedu.2019.103674

[20] Kasepalu, R., Prieto, L. P., Ley, T., & Chejara, P. (2022). Teacher Artificial Intelligence-Supported Pedagogical Actions in Collaborative Learning Coregulation: A Wizard-of-Oz Study. Frontiers in Education, 7. https://doi.org/10.3389/feduc.2022.736194

[21] Rolim, V., Ferreira, R., Lins, R. D., & Gǎsević, D. (2019). A network-based analytic approach to uncovering the relationship between social and cognitive presences in communities of inquiry. The Internet and Higher Education, 42, 53–65. https://doi.org/10.1016/j.iheduc.2019.05.001

[22] Praharaj, S.; Scheffel, M.;Schmitz, M.; Specht, M.; Drachsler, H. Towards Automatic Collaboration Analytics for Group Speech Data Using Learning Analytics. Sensors 2021, 21, 3156. https://doi.org/10.3390/s21093156

[23] David Williamson Shaffer and A. R. Ruis, Epistemic Network Analysis: A Worked Example of Theory-Based Learning Analytics, Wisconsin Center for Education Research, University of Wisconsin–Madison, USA DOI: 10.18608/hla17.015

[24] Gasevic, Dragan & Joksimovic, Srecko & Eagan, Brendan & Shaffer, David. (2018). SENS: Network analytics to combine social and cognitive perspectives of collaborative learning. Computers in Human Behavior. 92. 10.1016/j.chb.2018.07.003.

[25] Roschelle, J., Teasley, S.D. (1995). The Construction of Shared Knowledge in Collaborative Problem Solving. In: O'Malley, C. (eds) Computer Supported Collaborative Learning. NATO ASI Series, vol 128. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-85098-1_5

[26] Peter Guiney, Tertiary Sector Performance Analysis, Ministry of Education, Technology-supported physical learning spaces: An annotated bibliography,2016

[27] J. D. Walker; D. Christopher Brooks; Paul Baepler (2022). Pedagogy and Space: Empirical Research on New Learning Environments.. EDUCAUSE Quarterly 34. https://eric.ed.gov/

[28] S. A. Viswanathan and K. VanLehn, "Using the Tablet Gestures and Speech of Pairs of Students to Classify Their Collaboration," in IEEE Transactions on Learning Technologies, vol. 11, no. 2, pp. 230-242, 1 April-June 2018, doi: 10.1109/TLT.2017.2704099.

[29] Viswanathan, S. A., & VanLehn, K. (2019). Collaboration Detection that Preserves Privacy of Students' Speech. Artificial Intelligence in Education, 507–517. https://doi.org/10.1007/978-3-030-23204-7_42

[30] Rao, P. (2008). Audio Signal Processing. In: Prasad, B., Prasanna, S.R.M. (eds) Speech, Audio, Image and Biomedical Signal Processing using Neural Networks. Studies in Computational Intelligence, vol 83. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-75398-8_8

5. APPENDIX

In this section we have provided additional information which is important to understand the background and scope of the proposed work.

5.1 Definition of collaborative learning

Collaborative learning is a broad term applied to diverse learning situations. Inclusive definition of collaborative learning is as follows: "*It is a situation in which two or more people learn or attempt to learn something together*” [8]. This definition can be interpreted in many different ways. We created visual representation of the elements of the definition of collaborative learning as shown in figure 2.

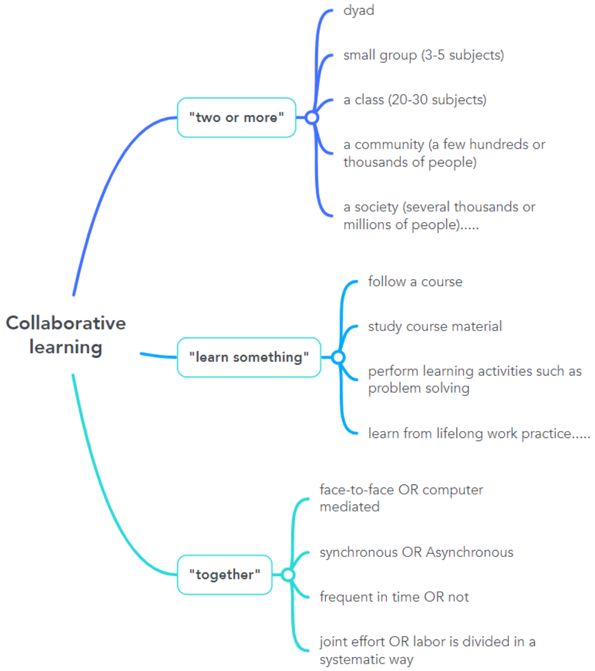

Collaborative learning
"two or more"
dyad
small group (3-5 subjects)
a class (20-30 subjects)
a community (a few hundreds or thousands of people)
a society (several thousands or millions of people).....
"learn something"
follow a course
study course material
perform learning activities such as problem solving
learn from lifelong work practice.....
"together"
face-to-face OR computer mediated
synchronous OR Asynchronous
frequent in time OR not
joint effort OR labor is divided in a systematic way

Figure 2. Definition of “Collaborative Learning”- visual representa-tion.

Another popular definition of collaboration is: *"... a coordinated, synchronous activity that is the result of a continued attempt to construct and maintain a shared conception of a problem"* [25]. The PISA 2015 collaborative framework defines collaboration as “*a process whereby two or more agents attempt to solve a problem by sharing the understanding and effort required to come to a solution, and pooling their knowledge, skills, and efforts to reach that solution*” [2].

5.2 Collaborative learning spaces

Technology-enhanced collaborative learning classroom used for conducting the study is shown in figure 3.

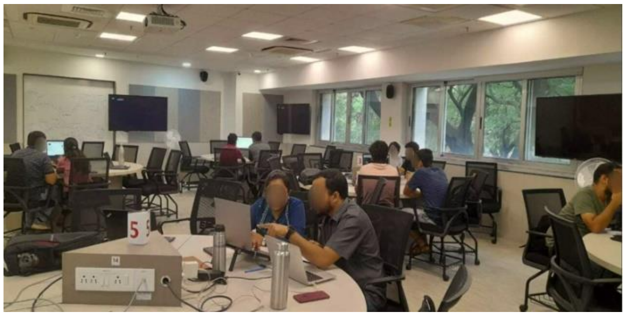


Figure 3. The technology-enhanced collaborative learning classroom.

The major benefits of creating technology-supported physical learning spaces are more frequent and higher quality teacher-student and student-student interactions, increased student usage of, and satisfaction with, the learning space, and authentic learning experiences [26]. It is also observed that, the type of space in which a class is taught influences instructor and student behavior in ways that likely moderate the effects of space on learning [27].

Analyzing the impact of metacognition prompts on learning in CBLE

Jyoti Shaha
Indian Institute of Technology Bombay
214380004@iitb.ac.in

Ramkumar Rajendran
Indian Institute of Technology Bombay
ramkumar.rajendran@iitb.ac.in

ABSTRACT

The computer-based learning environment (CBLE) is designed for instructional purposes and to support the learner to understand challenging and complex topics that are difficult to describe or comprehend. In CBLE, learners can access information in various formats such as text, diagrams, graphs, audio, video, etc. to learn. To successfully interact and learn from CBLE, the learners should plan their learning strategy, identify all the learning paths to achieve their learning goal, and select the most suitable one. However, navigating in such an environment can overwhelm learners' working memory, leading to cognitive overload, and disorientation which makes hurdles in learning. Several empirical studies have investigated overcoming the above challenges. They have reported, that the learners should be provided with metacognitive support. Metacognition is one of the strategies for encouraging self-regulated learning (SRL) in CBLEs. Hence, in our research, we propose to provide metacognitive prompts to learners while they interact with the CBLE and analyze the impact of metacognitive prompts.

Keywords

Metacognition, Metacognition prompts, Self-regulated learning, CBLE.

1. INTRODUCTION

The Computer-Based Learning Environment (CBLE) aims to support learners in achieving their objectives across a range of disciplines [1]. It incorporates multimedia, text, images, animations, simulations, audio, and video representations, among other things [6] for learners to access information [5]. Although CBLE provides excellent resources, it can also present challenges for learners. Since these environments give learners a high degree of control [6], they can follow their instructional path and access numerous representations of information as well as opportunities to manipulate them [5]. However, managing such an interactive and complex system actively can overwhelm learners' working memory, leading to cognitive overload, disorientation, and impeding the learning process [5]. Moreover, it has been reported that to acquire conceptual knowledge of a complex topic, learners should be able to constantly identify relevant information, track progress toward the goal, and sub-goals, and make judgments about their learning as per their learning progress [6] [9]. Recent studies have found that most learners are unable to manage their learning and they struggle to regulate multiple learning processes and as a result, learn less conceptually [1]. Students who can self-regulate their learning effectively are likely to acquire a conceptual understanding of complex topics [6]. Several empirical studies have investigated that to overcome the challenges provided by the environment, it is necessary to use metacognitive skills like monitoring, planning, and reflecting [5]. In order to engage in the planning, strategy usage, and monitoring processes, learners who do not self-initiate effective SRL processes should be assisted in identifying the metacognitive processes that are most effective for them [6]. Metacognitive support is one strategy for encouraging self-regulated learning [4]. The use of prompting as an instructional strategy is becoming more popular, particularly in the area of computer-based learning environments where prompting is simple to implement [6]. Several studies have revealed that metacognitive prompts direct the learners' awareness and monitor their learning activity which led to improvement in the planning, monitoring, and reflection activities in addition to learning outcomes [2] [7] [8]. In our study, we are intended to investigate the impact of metacognitive prompts on learning gain in CBLE. And also investigate possible factors that may have influenced the effectiveness of metacognitive prompts.

2. RESEARCH QUESTIONS

The focus of this study lies on metacognitive prompts, a topic that has been extensively investigated in the literature [3] [1] [4]. Drawing upon prior research metacognitive prompts can be categorized based on aspects such as modality, adaptability, and specificity. Mode of delivery is one such aspect, with prompts being classified as on-screen text, pop-up windows, virtual images, and auditory narration. Additionally, prompts can be tailored to the task at hand or learning situation, with adaptive prompts tailored to the individual needs of each student, while fixed prompts remain the same for all students. The effects of these prompts on metacognitive strategies and learning outcomes have been found to vary depending on the moderator variable [7]. However, most studies in this area have been conducted in the fields of social science (e.g., education, psychology) and science (e.g., math, biology, physics). Fewer studies have been conducted in the domain of engineering and technology, and even fewer have focused on problem-solving learning. While several studies have examined the impact of personalized metacognitive prompts and feedback on learning performance, there is still insufficient data on the performance of transfer and retention tasks, which would provide a clearer picture of the long-term effects of these prompts. Therefore, further research is needed to address some of the key research questions outlined below.

1. Do domain-specific, personalized metacognitive prompts with feedback help in enhancing the performance of transfer and retention tasks?

J. Shaha and R. Rajendran. Analyzing the impact of metacognition prompts on learning in cble. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 561–563, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115792

2. Do domain-specific, personalized metacognitive prompts with feedback help in enhancing metacognitive strategies?

3. PROPOSED CONTRIBUTION

To achieve our research goals, we propose to implement Design-based research (DBR) for the design and development of a CBLE aimed at promoting metacognition and improving learning performance among undergraduate engineering learners. Our study aims to investigate the impact of metacognitive prompts on learning gains in the CBLE and factors that may influence their effectiveness. The CBLE environment will feature the integration of concepts, practices, videos, text, simulations, and personalized prompts with feedback. A tentative plan outlining the research tasks to be undertaken is presented for further exploration.

Step 1 - Literature review in the context of undergraduate engineering classroom

The primary aim of this literature review was to examine the different interventions used to foster metacognition and different methodologies were used to measure the impact of the intervention on student performance. We identified primarily three intervention methods that were used to foster metacognition and train the purpose and strategies of metacognition like workshops, reading materials, and rubrics to guide learners. However, we identified three different methods to measure metacognitive awareness, reflection journals on their learning, metacognitive awareness questionnaires, and semi-structured interviews.

Step 2- Conduct a study to identify and assess the metacognitive awareness of undergraduate engineering learners

On the basis of the literature review, we conducted a research study with engineering students to understand the metacognitive process. We found that students mostly use control and regulation while using an open-ended learning environment. The low-scoring students often don't perform monitoring and reflection phases.

Step 3- Design and develop a system to foster engineering learners' metacognition

The proposed CBLE system is designed to help learners learn about electrical circuits. This CBLE environment will integrate concepts, practices, videos, text, simulations, and personalized prompts with feedback. The simulator will be designed to be user-friendly, with a variety of tools and features to help students build and analyze circuits. In addition to the simulator, the system will include a variety of video content that covers the key concepts and principles of electrical circuits. The videos will also provide step-by-step instructions on how to use the circuit simulator, so students can quickly get up to speed. It will also include text content that covers the same topics as the videos. The text content will be designed to be comprehensive and easy to understand, with clear explanations and examples. The text content will be organized into a list of topics to learn, so students can navigate and find the necessary information. The system will monitor the learner's progress and performance within the CBLE, and provide personalized metacognitive prompts to the learner when they are struggling or when they have made a mistake. Figure 1 shows our proposed study design and expected outcome

Step 4-Design the study to collect data and then analyze the data

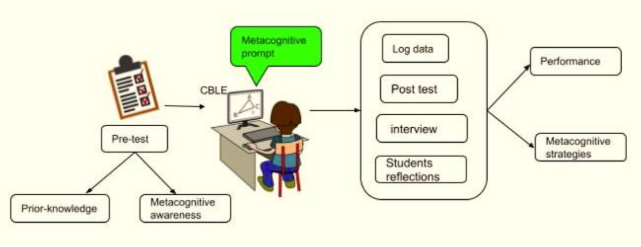


Figure 1. Proposed Study Design and expected outcomes

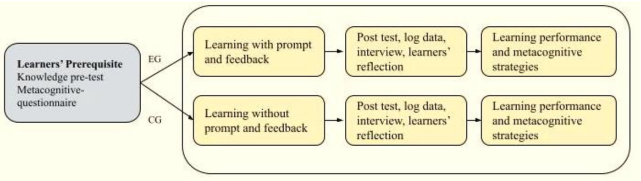

Learners' Prerequisite
Knowledge pre-test
Metacognitive-questionnaire
EG
CG
Learning with prompt and feedback
Learning without prompt and feedback
Post test, log data, interview, learners' reflection
Post test, log data, interview, learners' reflection
Learning performance and metacognitive strategies
Learning performance and metacognitive strategies

Figure 2. Study design and data collection

To address the research questions, we are targeting undergraduate students. Figure 2 shows the study design and data collection. The proposed CBLE system will have two versions, the Experimental Group (EG) will use the CBLE system with personalized metacognitive prompts and feedback, while the Control Group (CG) will use a CBLE system without personalized prompts and feedback. Participants will be randomly assigned to either experimental or control groups.

Data Collection: We plan to collect both groups' log data like learners' activity logs which track the actions taken by learners within the CBLE system, such as viewing content, completing exercises, and interacting with the simulation tools. And the time spent on each activity. This log data can provide insights into how learners engage with the system. Log data will be used to track the use of metacognitive prompts, these logs will provide insight into when and how often prompts are being used by students. Along with logs we have planned to collect pre-test, and post-test scores, and learners' reflections to analyze learners' performance. The pre-and post-tests will consist of multiple-choice questions and open-ended questions that assess students' understanding of the concepts covered in the CBLE system.

A delayed post-test will be administered a few weeks or months after the completion of the course to measure retention of learning. A transfer task can be administered to measure the extent to which students can apply what they have learned in a new context.

To analyze the learning strategies, the Motivated Strategies for Learning Questionnaire (MSLQ), a metacognitive questionnaire will be used. MSLQ can provide insights into the impact of personalized metacognitive prompts and feedback on students' learning strategies data and learners' reflections. Along with this, we will collect qualitative data through interviews.

Data Analysis: Data will be analyzed using both descriptive and inferential statistics. Descriptive statistics will be used to summarize the data and to identify any patterns or trends. Inferential statistics will be used to determine whether there are significant differences between the experimental and control groups in terms of student performance.

4. CONCLUSION

In conclusion, this research paper proposed the use of Design-based research (DBR) to design and develop a Computer-Based Learning Environment (CBLE) aimed at promoting metacognition and improving learning performance among undergraduate engineering learners. The study aimed to investigate the impact of metacognitive prompts on learning gains in the CBLE and factors that may influence their effectiveness. Based on a literature review, a study was conducted to identify and assess the metacognitive awareness of undergraduate engineering learners. A CBLE system was designed and developed, integrating concepts, practices, videos, text, simulations, and personalized prompts with feedback. A study design was proposed to collect data, including pre-tests, post-tests, and delayed post-tests, along with qualitative data through interviews. Data analysis will be conducted using both descriptive and inferential statistics. The expected outcomes of this research are to contribute to the understanding of the impact of metacognitive prompts on learning gains in a CBLE and factors that may influence their effectiveness. This research has implications for the design and development of effective CBLEs that can promote metacognition and improve learning outcomes.

5. REFERENCES

[1] Azevedo, R., Cromley, J. G., & Seibert, D. (2004). Does adaptive scaffolding facilitate students' ability to regulate their learning with hypermedia?. Contemporary educational psychology, 29(3), 344-370.

[2] Azevedo, R., Cromley, J. G., Moos, D. C., Greene, J. A., & Winters, F. I. (2011). Adaptive content and process scaffolding: A key to facilitating students' self-regulated learning with hypermedia. Psychological Test and Assessment Modeling, 53(1), 106.

[3] Bannert, M., Hildebrand, M., & Mengelkamp, C. (2009). Effects of a metacognitive support device in learning environments. Computers in human behavior, 25(4), 829-835.

[4] Engelmann, K., Bannert, M., & Melzner, N. (2021). Do self-created metacognitive prompts promote short-and long-term effects in computer-based learning environments?. Research and Practice in Technology Enhanced Learning, 16(1), 1-21.

[5] Winters, F. I., Greene, J. A., & Costich, C. M. (2008). Self-regulation of learning within computer-based learning environments: A critical analysis. Educational psychology review, 20, 429-444.

[6] Greene, J., Moos, D., & Azevedo, R. (2011). Self-regulation of learning with computer-based learning environments. New directions for teaching and learning. Publicado en línea en Wiley Online Library. https://doi. org/10.1002/tl, 449.

[7] Guo, L. (2022). Using metacognitive prompts to enhance self-regulated learning and learning outcomes: A meta-analysis of experimental studies in computer-based learning environments. Journal of Computer Assisted Learning, 38(3), 811-832.

[8] Pieger, E., & Bannert, M. (2018). Differential effects of students' self-directed metacognitive prompts. Computers in Human Behavior, 86, 165-173.

[9] Azevedo, R. (Ed.). (2018). Computers as Metacognitive Tools for Enhancing Learning: A Special Issue of Educational Psychologist. Routledge.

Designing a Learning Environment to Foster Critical Thinking

Ram Das Rai
Indian Institute of Technology Bombay
214383001@iitb.ac.in

ABSTRACT

In an era of digital media and hyperconnectivity, individuals are frequently flooded with an abundance of information, much of which they are unable to effectively process and are thus vulnerable to misinformation. One particularly harmful form of misinformation is the proliferation of "fake news," which can have a damaging effect on the social cohesion of communities. In response to this issue, educational practitioners in various nations are striving to empower learners with the ability to identify and refute such misinformation. The present author is also contributing to this effort through the development of a technology-enhanced learning environment that is intended to foster critical thinking in learners.

Keywords

fake news, digital literacy, critical thinking, learning environment

1. INTRODUCTION

In contemporary society, individuals are frequently overloaded with an abundance of information. However, this flood of information can often prove overwhelming, resulting in a vulnerability to misinformation. The internet serves as the primary means through which individuals access information. However, the knowledge readily available through this medium possesses distinct characteristics as compared to that traditionally provided by educators and educational texts [6]. Internet search results often comprise multiple accounts with varying scopes, arguments, and levels of support. Furthermore, online sources may vastly differ in terms of authorship, purpose, perspective, legitimacy, and justification techniques.

As previously discussed, the proliferation of misinformation in contemporary society has resulted in an increased risk of the formation of false beliefs among citizens. The inability to differentiate between legitimate and illegitimate information can lead to the acceptance of both as factual. To combat this, citizens must possess specialized skills that enable them to effectively navigate and evaluate the credibility and reliability of the vast amount of information available online. In light of this, there is a pressing need to understand and foster critical data literacy within the fields of educational research and practice. The objective of the present research is to facilitate the development of these skills among students, enabling them to proficiently analyze and scrutinize the reliability of complex online information.

2. BACKGROUND

In this background section, the author will first attempt to arrive at a working definition of critical thinking. In the next subsection, a number of organizations working to combat misinformation will be discussed. This will be followed by a discussion of academic research on misinformation and critical thinking in the next subsection. Finally, the author will discuss the current status of work in this area and how their own proposed research study will add to the existing knowledge.

2.1 Critical Thinking

Dwyer, Hogan, & Stewart define critical thinking as “a metacognitive process that, through purposeful, reflective judgment, increases the chances of producing a logical conclusion to an argument or solution to a problem” [7]. For the purpose of this paper, the author will use this definition as a working definition. However, it is necessary to also understand the broader meaning of critical thinking. To start with, Ennis has outlined abilities such as analyzing arguments, claims, or evidence, making inferences using inductive or deductive reasoning, judging or evaluating, and making decisions or solving problems as essential parts of critical thinking [8]. He has further identified behaviors relevant to critical thinking such as asking and answering questions for clarification; defining terms; identifying assumptions. Thus, critical thinking consists of a cluster of skills and behavior to analyze complex information. Looking at these aspects of critical thinking, it seems to be an effective tool for combating misinformation. However, fighting misinformation at an individual level is not enough. Fortunately, several organizations are also currently working on fighting misinformation, the details of their work are covered in the next section.

2.2 Misinformation Bunking Initiatives

Top universities and SMEs from seven different European nations are partners in the EU-funded initiative Co-Inform. The goal is to develop tools that promote digital literacy and critical thinking for a more informed society [5]. Their objective is to give individuals, journalists, and politicians the resources they need to recognize "fake news" online, comprehend how it spreads, and access reliable information. Co-Inform offers two main tools to combat misinformation. First is a browser plugin to increase citizens' awareness of content that is entirely or partially inaccurate, relevant fact-checking articles and remedial information, how ordinary citizens see this content, and important comments from fellow citizens that are both in favor of and against it. Second is a dashboard for fact-checking journalists and policymakers that displays discovered misinformation, its source, how and where it spreads and will spread in the future, the public's impression of it now and in the future, and the most important comments made by the public.

R. D. Rai. Designing a learning environment to foster critical thinking. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 564–568, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115681

There are also dedicated websites both at the global level and in India to track and debunk fake news and misinformation. Snopes, formerly known as the Urban Legends Reference Pages, is one such fact-checking website [20]. Snopes seeks to disprove or validate widely circulated urban legends. Similarly, Alt News is a fact-checking website based out of India that works to dispel the falsehoods, lies, and misinformation that people often come across in both mainstream and social media [2]. Politics, social media rumors, mainstream media misinformation, and bias are just a few examples of the inaccurate information that Alt News fact-checks. Apart from these initiatives for combating misinformation, there has also been a fair amount of academic research conducted in this field, the next section elaborates on that.

2.3 Digital Literacy Research

Digital literacy has been defined as the ability and knowledge required to effectively navigate the complex and fragmented world of information available online. In simpler terms, it means having the skills needed to find, understand, and use information on the internet. [9]. Three types of digital skills have been identified by Ng: technological (using technology tools); cognitive (using critical thinking while handling information); and social (communicating and socializing) [18]. In the context of the educational process, thinking skills have been recognized as an important component of digital literacy, along with technical abilities [13]. Sulzer asserts that digital thinking will include identifying misinformation, echo chambers, and fake news [21]. Thus, it can be argued that digital thinking is the term used by digital literacy practitioners to refer to critical thinking while engaging with information online.

Students engage with online information through their personal epistemology. This ability is related to their perspectives about knowledge and knowing. Kuhn and Park have characterized epistemological understanding at four levels [16]. At the first level, realists consider knowledge to be definite and to emanate from an outside source. They think it's not vital to use critical thinking. At the second level, absolutists think knowledge is certain and emanates from a distant source that is inaccessible. Thus, critical thinking serves as a means for people to evaluate claims in light of reality and decide whether they are true or not. Multiplists, who consider knowledge to be created by human minds and hence uncertain, are found at the third level. As a result, they believe that critical thinking is useless. The fourth level is where evaluativists reside, who think that knowledge is created by human minds, and is unclear, yet subject to review. They consider critical thinking as a tool for supporting reasonable claims and advancing understanding. In order to better understand the relationship between students' individual epistemologies and their online learning practices, Barzilai and Zohar studied 38 sixth-graders [4]. The results demonstrate the significance of epistemic thinking in online inquiry learning. Students who were more familiar with evaluation strategies and criteria performed more frequent and thorough website evaluations. Students who were more conscious of the potential for discrepancies between online accounts and the necessity of constructing knowledge by integrating several viewpoints were more likely to identify discrepancies between the points of view of various websites, compare them, and build an argument based on a variety of online sources. Even though personal epistemology can account for a fair portion of the processes people use to process information, it also involves other factors like cognitive biases and epistemic emotions. Fig 1 explains the various factors involved in this process.

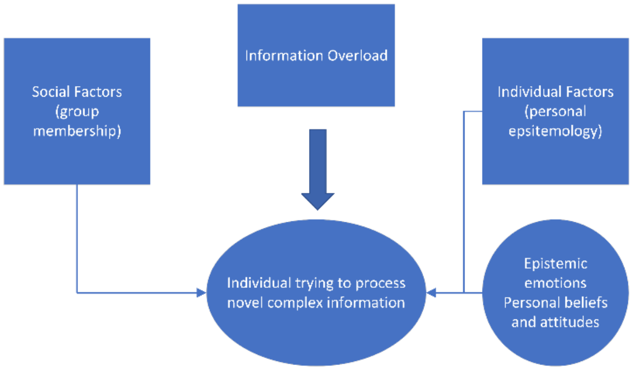

Social Factors (group membership)
Information Overload
Individual Factors (personal epsitemology)
Individual trying to process novel complex information
Epistemic emotions Personal beliefs and attitudes

Figure 1. Factors affecting complex information processing in individuals

Different research groups over the years have tried to design strategies to teach students how to use critical thinking to spot fake news. The Association of College and Research Libraries (ACRL) introduced its Framework for Information Literacy for Higher Education in 2015, which was officially adopted by the ACRL Board in January of the following year. This framework emphasizes the importance of learners utilizing research tools and evaluating the credibility of sources in order to develop their information literacy skills [3]. One tool that can assist students in this process is the use of LibGuides, which are web-based applications that allow for the creation and organization of electronic guides. These can be easily embedded in course and library websites and accessed by students online. An example of a useful LibGuide in this context is the "Fake" News guide created by librarian Eric Novotny in 2017 at The Pennsylvania State University. The guide covers various forms of fake news, such as satire, bias, and clickbait. Another helpful tool is the use of worksheets, such as the CRAAP (Currency, Relevance, Authority, Accuracy, and Purpose) worksheet developed by California State University, Chico in 2010. In contrast, The Global Digital Citizen Foundation (2015) promotes a different approach to critical thinking by using a "Who, What, Where, When, Why, and How" method.

In spite of the above efforts, a recent intervention study focused on teaching students to evaluate search results and select websites to open revealed that students frequently resorted to less effective tactics when analyzing results, using their familiarity with a website and its top-level domain to determine its reliability, despite teachers' best efforts to teach them strategies for evaluating results modeled on fact checkers' approaches [17]. This means that just informing students about strategies is not enough. Critical thinking is not just a bunch of skills but also an attitude which can only be inculcated through practice. A better strategy would be to provide opportunities to students to practice critical thinking abilities in context of actual problems in a learning environment. In this environment, they can practice their critical thinking abilities for long durations and hone them over time with proper support. Agesilaou & Kyza designed and implemented a Learning Environment to foster critical data literacy [1]. Their work describes the design-based research process of designing an educational intervention to foster critical data literacy through the use of self-tracking devices. While their work focused on the issues of privacy and digital data, the author of this current paper plans to design a learning environment to foster critical thinking skills among students in order to specifically deal with misinformation present on the internet. The research goal will be to help students to develop effective critical thinking strategies to deal dealing with complex online information.

This background section helped define critical thinking as a metacognitive process that involves analyzing arguments, claims, and evidence, making inferences, solving problems and asking questions for clarification and identifying assumptions. It also helped to understand the work being done in both academic and non-academic spheres to counter the spread of misinformation online. It explained how the information transfer techniques used by teachers fall short in equipping students with critical thinking abilities and established the need for a learning environment which offers prolonged opportunities for students to practice and hone their critical thinking skills on authentic scenarios where they are also provided continuous support in terms of active scaffolds. In the next section, the author will elaborate on their proposed plan for the learning environment.

3. PROPOSED SOLUTION

In order to build a coherent and credible learning environment to foster critical thinking, the author needs to use some theoretical perspectives to provide the foundation for such an intervention. One prominent theory is the dual processing theory, which proposes that individuals use two primary modes of thinking - System 1, which is intuitive and automatic, and System 2, which is more analytical and deliberate [10]. This theory is particularly relevant for this intervention design because it has been observed that many a times users fall prey to misinformation because they have not spent enough time on a piece of information and respond too quickly [11]. The author is aware that dual processing theory has faced criticism in recent years. However, there is still substantial evidence in cognitive science to support the dual-processing distinction [12]. The theory remains valid and useful in understanding the interplay between automatic and deliberate thinking processes.

In order to provide an authentic learning experience to the students, a problem-based learning (PBL) approach can be used, where students work in pairs to develop solutions to real-world problems related to misinformation [14]. In order to ensure a smooth collaboration between the student pairs, collaboration scripts will be used. Collaboration scripts are instructional tools that guide learners on how to interact with each other during learning activities [15]. They provide a sequence of learning activities and roles for learners to follow in order to promote collaborative learning. In this learning environment, the author plans to use a type of driver-navigator script where one partner searches for the information on the system while the other person guides them. The main motive behind this peculiar pairing is that it will require the learners to discuss and debate the entire time while they are searching for credible information because they will have their biases and beliefs.

A theory that can be leveraged to better understand this social interaction is the social judgment theory, which suggests that people's attitudes and beliefs are influenced by their perception of what others think [19]. By encouraging students to engage in discussions and debates with their peers, the learning environment can foster critical evaluation of information by considering multiple perspectives and identifying potential biases. Introducing the social learning aspect is particularly important for the students to learn to argue logically and identify fallacies and biases in other's and their own opinions.

Also, as discussed in the background section above, different learners tend to use different ways of engaging with information online. There is also this idea of the behavioral pattern displayed by professional fact-checkers when trying to determine the authenticity of a piece of online media. The best way to measure these various techniques will be to use learning analytics to capture the interaction of learners with the learning environment using log data. Primarily, two kinds of interactions will be captured, first will be the frequency of user interaction with various com-ponents in the environment, and second will be the duration of those interactions. This will help the researcher categorize the various patterns, for example, one cohort of users might be clicking on a number of resources and spending little time on each of them while another cohort might be accessing only few resources but spend significant time on each resource. Later, the various cohorts would be analyzed in relation to their demonstration of critical thinking behaviors. In the next sub-section, the author will provide a sample learning task that they might use in their learning environment.

3.1 Sample Learning Problem

This scenario is designed to help students practice critical thinking skills in the context of evaluating claims made about a dietary supplement marketed as a weight loss aid. Two students as a pair will work to evaluate the claims made about the supplement and pronounce their verdict on whether it is a weight loss solution or not. They will also be asked to back up their verdict with proper evidence. The students will be given the following sources of information:

1. An advertisement that claims the supplement is a "miracle weight loss solution" and features testimonials from people who have lost weight while taking the supplement.
2. A medical study that reports on the potential health risks associated with the supplement, like organ damage and other serious side effects.
3. A warning from a (government) health agency that advises consumers to avoid the supplement due to its potential health risks.

To help students evaluate the information provided, they will be given the following questions:

1. What are the claims being made about the effectiveness of the dietary supplement as a weight loss aid?
2. What evidence is provided to support these claims, and how strong is this evidence?
3. What are the potential health risks associated with the supplement, and how serious are these risks?
4. What are the recommendations of (government) health agencies with regards to the supplement?
5. What are the potential biases or conflicts of interest that may be present in each source of information?

This scenario is designed to help students develop critical thinking skills related to evaluating the claims made about dietary supplements and to identify potential biases or conflicts of interest in the sources of information provided. This is just a single sample problem and similar other problems from socio-scientific domain would be developed to be used in the learning environment.

4. RESEARCH QUESTIONS

The study will attempt to answer these four primary research questions.

1. To what extent does training in critical thinking skills help students identify fake news more effectively?
2. To what extent does collaborative learning help improve critical thinking skills among students while processing complex online information?

3. What are the various categories of learners in terms of their behavior while processing complex online information?
4. How do these various categories of learners differ in terms of use of critical thinking skills?

5. METHOD

5.1 Methodology

This research study will use a mixed-method approach with a heavy tilt towards the qualitative side. To answer the first two research questions, qualitative data would be required and this is the primary focus of this study. In order to understand the learner behavior and answer the last two research questions, a quantitative approach will be used which will employ learning analytics technique.

5.2 Target population and sampling

Even though digital literacy is a skill that is helpful at all ages of life, this study will be conducted primarily amongst undergraduate and postgraduate program students as individuals at this level are young adults and misinformation can lead them to take faulty steps at this critical juncture of life. Based on this misinformation, they might develop faulty beliefs which might get stay with them throughout their life.

5.3 Data Collection Tools

For the purpose of this study, multiple data sources will be utilized. While the students work in groups, the audio and video of their discussions will be recorded, along with screen recordings of their system and any notes they have created during the discussion. This will be followed by a follow-up interview to further probe their epistemic strategies.

The log data of the students while they interact with the system will also be collected. As explained in the proposed solution section, the log data from the system will contain time stamps which correspond to specific user activities in the system. This log data can be exported in the form of an excel data sheet which can be further processed to find user behavior patterns.

Finally, in order to measure the effectiveness of the learning environment on the critical thinking abilities of the students, a pre and posttest will be conducted. This test will present learners with various scenarios involving fake news.

5.4 Data Analysis

A rigorous and systematic methodology in the form of inductive coding strategy will be implemented in order to systematically examine and interpret the qualitative data that has been meticulously gathered through the various data collection tools. This approach will involve the identification of meaningful patterns, themes, and categories within the data, in order to uncover underlying meaning and to gain a comprehensive understanding of effectiveness of critical thinking in identifying fake news. The utilization of an inductive coding strategy will allow for the development of an inductive theory that is grounded in the data, and which can offer insight into the complex phenomena of critical thinking and fake news identification.

The log data, comprising of time-stamped information, will be subjected to a process of sequential pattern mining. This process will enable the identification of recurring patterns and sequences within the data, and the generation of clusters of users who exhibit similar patterns of behavior within the learning environment. By utilizing this approach, it will be possible to gain a deeper understanding of the usage patterns and behaviors of the users within the learning environment, which in turn can be used to understand the relation of certain behavioral patterns to demonstration of various critical thinking levels. Thus, the results generated by the sequential pattern mining algorithm will be used to model the behavior of users in terms of their display of critical thinking skills. This user behavior can be compared to the patterns displayed by professional fact checkers. In this way, desirable patterns of use of critical thinking skills can be identified which can be used for improving the critical thinking training aspects of the learning environment. This data can also be used to test new users and predict what kind of learning interventions would be required to help them develop necessary critical thinking abilities.

6. DISSERTATION STATUS AND NEXT STEPS

The current research is a continuation of the work being done at the author’s organization in the field of technology enhanced learning of thinking skills. The organization in the past has conducted numerous studies ranging on various thinking skills from historical thinking, design thinking to estimation and more. There are also ongoing research projects that explore certain thinking skills like systems thinking. This current thesis research is a continuation of this work in terms of addressing more social problems like fake news and misinformation. This work draws from and builds on previous work done in the form of question posing and hypothesis testing skills amongst students.

The current author is currently in the early years of their PhD research and so the research plan is in in its nascent stage. Thus, a major reason for writing this paper is also to get helpful guidance from the members of the research community. Currently, the author has two immediate tasks in front of them. First, they plan to conduct a thorough literature review of critical thinking as a digital literacy skill, particularly in the educational context. The second task is to conduct a preliminary research study to explore how social interaction affects students' epistemic thinking in online inquiry learning. The author plans to use the following conjecture for this preliminary study: when working in groups, students' personal epistemologies will interact with each other and help to reflect on each other's cognitive biases and epistemic emotions.

7. EXPECTED CONTRIBUTIONS

The present study is situated in a broad context of combating online fake news and equipping citizens with skills to process information overload. However, the novel contribution of this study will be to explore the effects of social interaction, in terms of collaborative work, on critical thinking abilities of students. This line of argument is rooted in the concept of democracy where vigilant citizens hold each other accountable in terms of their beliefs and practices. The research also has more immediate contributions in terms of understanding how different groups of people employ critical thinking while processing complex information online. This can help to create better resources for supporting people in spotting misinformation and debunking fake news. This whole process will affect society on two levels, at individual levels, people can become more conscious of their own biases and logical errors and a at a societal level, people can have more fruitful conversations across different thought camps as they will have a more solid ground of information to engage in discussion.

8. ASPECTS OF THE RESEARCH ON WHICH ADVICE IS SOUGHT

Since the author is in their preliminary stage of research, they would be open to suggestions on almost every aspect of the study. However, the author is particularly interested in discovering more effective ways to capture critical thinking behavior of participants. As of now, the author is using a mixed-method approach and is relying on log data in terms of frequency and duration for capturing interaction. The author would like to receive more suggestions on how this aspect can be made more effective.

9. ACKNOWLEDGMENTS

The author expresses their sincere gratitude to Prof. Sahana Murthy and Prof. Ramkumar Rajendran from IIT Bombay for their invaluable guidance and support.

10. REFERENCES

[1] Agesilaou, A., & Kyza, E. A. (2021). Empowering Students to be Data Literate: The Design and Implementation of a Learning Environment to Foster Critical Data Literacy. In *Proceedings of the 15th International Conference of the Learning Sciences-ICLS 2021*. (pp. 458-465). International Society of the Learning Sciences.

[2] Alt News. (2021, June 11). About. Retrieved September 19, 2022, from https://www.altnews.in/about/

[3] Association of College and Research Libraries. 2017. “Framework for Information Literacy for Higher Education.” Accessed March 4, 2017. http://acrl.ala.org/framework/

[4] Barzilai, S., & Zohar, A. (2012). Epistemic thinking in action: Evaluating and integrating online sources. *Cognition and Instruction*, 30(1), 39-85.

[5] Co-inform. (n.d.). The Project. Retrieved September 19, 2022, from https://coinform.eu/about/the-project/

[6] Dede, C. (2008). A seismic shift in epistemology. *Educause Review*, 43, 80–81

[7] Dwyer, C. P., Hogan, M. J., & Stewart, I. (2014). An integrated critical thinking framework for the 21st century. *Thinking skills and Creativity*, 12, 43-52.

[8] Ennis, R. H. (1989). Critical thinking and subject specificity: Clarification and needed research. *Educational researcher*, 18(3), 4-10.

[9] Eshet, Y. (2004). Digital literacy: A conceptual framework for survival skills in the digital era. *Journal of Educational Multimedia and Hypermedia*, 13(1), 93–106, https://www.learntechlib.org/primary/p/4793/

[10] Evans, J. S. B. (2003). In two minds: dual-process ac-counts of reasoning. Trends in cognitive sciences, 7(10), 454-459.

[11] Evans, J. S. B., & Curtis-Holmes, J. (2005). Rapid responding increases belief bias: Evidence for the dual-process theory of reasoning. Thinking & Reasoning, 11(4), 382-389.

[12] Evans, J. S. B., & Stanovich, K. E. (2013). Dual-process theories of higher cognition: Advancing the debate. Perspectives on psychological science, 8(3), 223-241.

[13] Ferrari, A. (2012). Digital competence in practice: An analysis of frameworks. *JCR IPTS*, Sevilla. https://ifap.ru/library/book522.pdf

[14] Hmelo-Silver, C. E. (2004). Problem-based learning: What and how do students learn?. Educational psychology re-view, 16, 235-266

[15] Kobbe, L., Weinberger, A., Dillenbourg, P., Harrer, A., Hämäläinen, R., Häkkinen, P., & Fischer, F. (2007). Specifying computer-supported collaboration scripts. International Journal of Computer-Supported Collaborative Learning, 2, 211-224

[16] Kuhn, D., & Park, S. H. (2005). Epistemological understanding and the development of intellectual values. *International Journal of educational research*, 43(3), 111-124.

[17] McGrew, S., & Glass, A. C. (2021). Click Restraint: Teaching Students to Analyze Search Results. In *Proceedings of the 14th International Conference on Computer-Supported Collaborative Learning-CSCL 2021*. (pp. 145-148). International Society of the Learning Sciences.

[18] Ng, W. (2012). Can we teach digital natives digital literacy? *Computers & Education*, 59(3), 1065–1078. https://doi.org/10.1016/j.compedu.2012.04.016

[19] O’Keefe, D. J. (2016). Social Judgment Theory. In Persuasion: Theory and research (III, pp. 48–70). New Delhi: Sage Publications

[20] Snopes. (2021, August 3). About Us. Retrieved September 19, 2022, from https://www.snopes.com/about/

[21] Sulzer, A. (2018). (Re)conceptualizing digital literacies before and after the election of Trump. *English Teaching: Practice & Critique*, 17(2), 58–71. https://doi.org/10.1108/ETPC-06-2017-009

Response Process Data in Educational and Psychological Assessment: A Scoping Review of Empirical Studies

Guanyu Chen
The University of British Columbia
cguanyu@student.ubc.ca

Yan Liu
Carleton University
YanLiu5@cunet.carleton.ca

ABSTRACT

With the advance of computerized assessments, response process data (RPD) become available. RPD has been increasingly gaining popularity because it can help to understand and study the cognitive processes of test takers. We aim to conduct a scoping review to provide a comprehensive overview of the common practice and major findings with a focus on the theoretical framework and analytical methods applied in RPD studies. This review can help researchers understand the advantages and challenges of using RPD in both educational and psychological fields. Our findings provide guidance to researchers who are interested in RPD applications.

Keywords

scoping review, response process data, log file data, assessment

1. INTRODUCTION

With the recent development of computer technology, response process data (RPD) are widely collected in computerized assessments [1]. RPD reflect the thinking processes, strategies, and behaviors of test takers when they read, interpret, and formulate solutions to assessment tasks [2]. RPD can document test-taking behaviors that may not be observed directly from test scores, which can show response patterns and thinking processes and may possibly provide learners and other stakeholders with more meaningful feedback [3]. Given the importance of RPD, the purpose of this scoping review is to examine the extent, range, and characteristics of RPD, to summarize analytical methods used as well as the findings obtained from application studies, and to identify gaps in the literature [4].

2. RESPONSE PROCESS DATA

RPD can be traced back to the log files, which record events that occur in a computer system [5]. RPD is one type of log-file data, also known as (response-related) paradata in survey research [6], recording the interactions between the test takers and the computer [7]. In computer-based assessment contexts, both the test takers' actions to the stimulus materials and the ordered sequence (i.e., the timestamps) of these actions are stored in RPD [2], [3], [8]. RPD are usually stored in a structured format, such as XML and JSON, and RPD need to be parsed and converted into a tabular data frame for further analysis [9]. Table 1 is an adapted example from a problem-solving task in Programme for International Student Assessment (PISA) 2012. The first column contains the type of event, including system-generated events (start item, end item) and student-generated events (e.g., ACER_EVENT, click). The second column records the event time, given in seconds from the beginning of the assessment. The third column is the event sequence number. The fourth column provided detailed information (i.e., properties) about the event.

Table 1. Process Data from PISA 2012 Problem Solving

event	time	event_number	event_value
START_ITEM	0.10	1	NULL
ACER_EVENT	43.40	2	'00000000000010000000000
click	43.40	3	hit_nowhereSakharov
ACER_EVENT	44.90	4	'00000000000000000000000

3. STUDY PURPOSES

3.1 Scoping Review

Although RPD is an emerging topic and there are a number of empirical studies that have been conducted, there is no review that has been carried out to offer insights into the current applications related to RPD according to our best knowledge. A scoping review maps the key concepts behind a research topic and different sources of evidence, and the scoping review can be conducted as a standalone study, especially for a complex and emerging topic [10]. Conducting a scoping review will contribute to an overall understanding of the current application of RPD across different research areas in educational and psychological assessment.

Specifically, we will undertake the scoping reviews for examining the extent and characteristics of research with RPD. It is important to gain insights into how RPD are being applied and analyzed as a gold mine in educational and psychological assessment. By summarizing the current research, theoretical and analytical frameworks for RPD will be identified and examined for providing a broader overview of these indicators, methods, and findings. Finally, this scoping review could also be used to guide further research and practice.

3.2 Review Objects

RPD present a challenge to researchers as the underlying cognitive mechanisms of test takers are not always clear. Additionally, the format of RPD is not consistent with the traditional data format utilized in psychology and education, and analyzing them involves

G. Chen and Y. Liu. Response process data in educational and psychological assessment: A scoping review of empirical studies. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 569–571, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115660

managing a large volume of raw data without an established methodology to generate meaningful variables. Consequently, this scoping review aims to explore the use of RPD in educational and psychological assessments, taking into account the theoretical framework, meaningful indicators, and analytical methods employed. Specifically, this review addresses four key questions:

(1) What theoretical frameworks are used in the analysis of RPD?

(2) How are suitable indicators extracted and generated from raw RPD, and what kinds of indicators have been utilized in current practice?

(3) What analytical methods have been employed in the study of RPD?

(4) Based on the indicators and corresponding methods utilized in existing studies, what inferences have been made, and what are the associated research purposes and findings?

4. METHODS

4.1 Study Design

We adapted Arksey and O'Malley's framework [10] and the Preferred Reporting Items for Systematic Reviews and Meta-Analyses Extension for Scoping Reviews (PRISMA-ScR) [4], [11] for conducting a scoping review. The six-stage framework proposed by Arksey and O'Malley includes identifying the research question, identifying relevant studies, study selection, charting the data, collating, summarizing and reporting the results, and consultation exercise. This framework helps ensure that the review is conducted in a comprehensive and transparent manner, providing a structured approach to identifying relevant studies and synthesizing the findings. After conducting the scoping review, PRISMA-ScR provides a standardized checklist for reporting scoping reviews, ensuring that important information is included in the final report. Following PRISMA-ScR guidelines helps ensure that the review is conducted in a rigorous and transparent manner, making it easier for readers to evaluate the validity and reliability of the study.

In sum, using Arksey and O'Malley's framework and PRISMA-ScR for scoping reviews is important as they provide a structured approach to identifying and synthesizing relevant literature and ensure that increases the validity and reliability of the study findings and makes it easier for readers to evaluate the review.

4.2 Search Strategy

The search query for the Web of Science was provided here with the consideration of the Peer Review of Electronic Search Strategies (PRESS) checklist [12]:

TS = ((paradata OR "process data" OR "log data" OR "log-file data" OR "logfile data" OR "mouse click" OR keystroke OR keypress) AND (survey OR questionnaire OR "test batter*" OR assessment OR PISA OR PIAAC OR NAEP OR TIMSS OR PIRLS)) AND PY = (2000-2022)

The query returned 1904 records in Web of Science and around 5000 records from all databases, including ERIC, Education Source, PsycInfo, ProQuest Dissertations & Theses Global, Web of Science, and Scopus.

4.3 Inclusion and Exclusion Criteria

As a scoping review, we include all types of empirical research, including gray literature from ProQuest Dissertations & Theses Global. Most of the studies use RPD as a secondary data analysis. Thus, we expect very few experimental studies, and most of the studies will be observational studies. As mentioned in Research Question Section, we focused on the theoretical and analytical frameworks of RPD in practice. Hence, we excluded methodological studies which focused on the simulation or algorithm. Review studies will be considered, and the empirical studies included in review studies will be retrieved and reviewed. However, we can only include full-text and English articles for conducting the full-text review according to the background of reviewers. Finally, this scoping review includes all human populations in any context as long as their interactions with computers were recorded.

4.4 Study Selection

Study selection is an iterative, rather than linear, stage involving a process of searching the literature, refining the search strategy, and reviewing articles for study inclusion and exclusion criteria [13]. At least two independent reviewers were asked to perform the study selection for the title and abstract screening and full-text screening. Another content expert was invited to solve the disagreement between the reviewers. Some pilot tests were recommended before the formal selection for refining this study selection process [11]. We will choose a sample of 50 articles, review these articles with eligibility criteria, discuss the discrepancies, and modify the search query and eligibility criteria.

We will use a flowchart of the review process from PRISMA-ScR to describe the whole scoping review process, including the databases, duplications, screening, full-text retrieval, and additional search from reference lists and relevant organizations. Covidence will be used for data management and screening.

5. EXPECTED RESULTS

5.1 Data Extraction

Google Forms will be used for developing the data charting form to collect the information for answering research questions. A series of key information will be recorded, such as:

(1) Citation information: author(s), publication year

(2) Indicators: generation, definition, type, theoretical framework

(3) Methods: name, category

(4) Inferential framework: aim of the study, findings

Note that additional information will be included during the review, and the chart form will be continually updated. After the review team discusses and trials the chart form and the chart form, two independent reviewers will extract the information to ensure the accuracy of data extraction.

5.2 Data Synthesis

To clarify our results, we will break our data synthesis into three steps [13]. First, we need to conduct the data analysis. The frequency counts of indicators, methods, and findings are used for depicting the extent, range, and characteristics of the studies included in the scoping review [10], [14]. Moreover, to provide in-depth analyses, descriptive qualitative data analysis, such as thematic analysis with human coding [15], will be used [11]. Thematic analysis can summarize the data into a particular category (i.e., classifying the statistical methods into descriptive statistics or inferential statistics). Then, according to the research questions, we will report the results and produce the findings. A small table includes the characteristics of all the studies under a specific topic, (i.e., indicators, methods, and inferential frameworks in this review). Finally, the implications of our results will be considered with the overall research purpose and the specific research question and extended to the broader context for future research, policy, and practice [13].

6. DISCUSSION

RPD is an emerging and developing research topic in the fields of psychology and education. With the wide use of computer-based assessment, RPD becomes more and more available. However, the significantly increased volume, velocity, and variety of RPD raise new challenges for researchers to handle, analyze, and interpret them in order to materialize the value [1]. As there is a lack of scoping review to provide a comprehensive overview of the current theoretical and analytical frameworks to guide future research and practice. Even though a variety of analytic methods were used for different indicators, this scoping review will provide a systematic summary of common indicators, methods, and findings.

7. REFERENCES

[1] von Davier, A. A., Mislevy, R. J., & Hao, J. (Eds.). (2021). *Computational Psychometrics: New Methodologies for a New Generation of Digital Learning and Assessment:* With Examples in R and Python. Springer International Publishing. https://doi.org/10.1007/978-3-030-74394-9

[2] Ercikan, K., & Pellegrino, J. W. (2017). *Validation of score meaning for the next generation of assessments: The use of response processes.* Taylor & Francis.

[3] Jiao, H., He, Q., & Veldkamp, B. P. (2021). Editorial: Process Data in Educational and Psychological Measurement. *Frontiers in Psychology*, 12. https://www.frontiersin.org/articles/10.3389/fpsyg.2021.793399

[4] Tricco, A. C., Lillie, E., Zarin, W., O'Brien, K. K., Colquhoun, H., Levac, D., Moher, D., Peters, M. D. J., Horsley, T., Weeks, L., Hempel, S., Akl, E. A., Chang, C., McGowan, J., Stewart, L., Hartling, L., Aldcroft, A., Wilson, M. G., Garritty, C., … Straus, S. E. (2018). PRISMA Extension for Scoping Reviews (PRISMA-ScR): Checklist and Explanation. *Annals of Internal Medicine*, 169(7), 467–473. https://doi.org/10.7326/M18-0850

[5] Andrews, J. H. (1998). Testing using log file analysis: Tools, methods, and issues. *Proceedings 13th IEEE International Conference on Automated Software Engineering (Cat. No.98EX239)*, 157–166. https://doi.org/10.1109/ASE.1998.732614

[6] Kroehne, U., & Goldhammer, F. (2018). How to conceptualize, represent, and analyze log data from technology-based assessments? A generic framework and an application to questionnaire items. *Behaviormetrika*, 45(2), 527–563. https://doi.org/10.1007/s41237-018-0063-y

[7] OECD. (2019). *Beyond Proficiency: Using Log Files to Understand Respondent Behaviour in the Survey of Adult Skills*. OECD. https://doi.org/10.1787/0b1414ed-en

[8] OECD. (2015). *Students, Computers and Learning: Making the Connection*. OECD. https://doi.org/10.1787/9789264239555-en

[9] Hao, J., & Mislevy, R. J. (2021). A Data Science Perspective on Computational Psychometrics. In A. A. von Davier, R. J. Mislevy, & J. Hao (Eds.), *Computational Psychometrics: New Methodologies for a New Generation of Digital Learning and Assessment* (pp. 133–158). Springer.

[10] Arksey, H., & O'Malley, L. (2005). Scoping studies: Towards a methodological framework. *International Journal of Social Research Methodology*, 8(1), 19–32. https://doi.org/10.1080/1364557032000119616

[11] Peters, M. D. J., Marnie, C., Tricco, A. C., Pollock, D., Munn, Z., Alexander, L., McInerney, P., Godfrey, C. M., & Khalil, H. (2020). Updated methodological guidance for the conduct of scoping reviews. *JBI Evidence Synthesis*, 18(10), 2119. https://doi.org/10.11124/JBIES-20-00167

[12] McGowan, J., Sampson, M., & Lefebvre, C. (2010). An Evidence Based Checklist for the Peer Review of Electronic Search Strategies (PRESS EBC). *Evidence Based Library and Information Practice*, 5(1), Article 1. https://doi.org/10.18438/B8SG8R

[13] Levac, D., Colquhoun, H., & O'Brien, K. K. (2010). Scoping studies: Advancing the methodology. *Implementation Science*, 5(1), 69. https://doi.org/10.1186/1748-5908-5-69

[14] Peters, M. D. J., Godfrey, C., McInerney, P., Munn, Z., Tricco, A. C., & Khalil, H. (2020). Chapter 11: Scoping reviews (2020 version). In A. E. & M. Z. (Eds.), *JBI Manual for Evidence Synthesis (Vol. 2020)*. JBI. https://doi.org/10.46658/JBIMES-20-12

[15] Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77–101.

Understanding Learners’ Alternative Conceptions through Interaction Patterns During analogical reasoning

Meera Pawar
Indian Institute of Technology, Bombay
214380002@iitb.ac.in

Sahana Murthy
Indian Institute of Technology, Bombay
sahanamurthy@iitb.ac.in

ABSTRACT

The success of vaccine development and distribution has highlighted the importance of immunology as a practical and relevant science. However, studying immunology can be challenging for college students as it requires them to engage with new vocabulary and advanced biological concepts. Due to this, many learners find it difficult to integrate the new material with their prior knowledge and lose interest in the subject when taught through a traditional didactic approach, which can decrease their engagement in the class. Lack of engagement and misinterpretation of instructions are a few of the reasons why learners develop alternative conceptions adding to their learning difficulties. There are various methods a teacher might use in a classroom to identify these alternative conceptions. Analogical reasoning is one such method to understand learners’ alternative conceptions. Along with learners’ reasoning, log data of learners’ interactions can provide insights that can be useful to model learners’ behavior. This will enable us to scaffold their learning. This study proposes the development of a technology-enhanced learning environment based on the theories of analogical reasoning through which learners’ interaction patterns can be captured and studied with the help of different data sets.

Keywords

Immunology, Human Immune System alternative conception, Analogical reasoning, interaction pattern, log data, Eye gaze

1. INTRODUCTION

Biological education consists of learning many complex systems which are integrated into each other. Learners as well as teachers struggle while learning and teaching respectively about different systems, components, functions, and mechanisms. Human Immunology is one of such complex biological systems made up of subsystems like the lymphatic system, blood cells, and antibodies. This system is divided into three components 1) the First Line of Defense 2) the Innate Immune system 3) the Adaptive Immune system. The human immune system works at different levels of the organization including cells, organs, tissues, and symptoms at the organism level. Traditional classroom method of teaching typically focuses on the transmission of information, due to which learners face a challenge in the understanding of human immunology. The processes and relationships between many lines of defense are more abstract than other biology topics, and learners typically lack the necessary background knowledge when they first meet the field [3]. Sometimes they misunderstand the concepts [4]. Alternative conceptions and misconceptions about human biology pose a serious challenge to medical education's emphasis on precise scientific and clinical reasoning

For enhanced comprehension, improved critical thinking, and learner engagement, various pedagogical techniques have been adopted, including case-based learning, team-based learning, and learning through tales and games [8]. Numerous studies demonstrate the usage of simple real-life analogies and metaphors that map to abstract concepts of immunology. In general, a comparison of two objects, or systems of objects, that focuses on the similarities between them is called an analogy [1]. Analogies have been used in many studies to address learners’ misconceptions and alternative conceptions. Analogies have been largely used in topics such as protein synthesis, the nervous system, and the immune system. [2]. Analysis of analogical reasoning to understand alternative conceptions has been done using rubrics, frameworks, and interviews [9]. The majority of current research uses multiple-choice pre- and post-tests along with qualitative data to monitor participants' performance or comprehension [10]. However, we would like to triangulate our research investigations using eye gaze data and log data of the interaction. To do this, we have proposed a learning environment based on the analogical process model framework [7]. In this environment, we have designed certain activities which would capture learners’ interaction log data. The learning environment will enable learners to interact with different components such as “Stage 1” where the learner will go through a reading task. At this stage eye gaze data will be collected as research in reading can contribute to our knowledge of how learners interpret the educational text. At “Stage 2” we will collect log data to interpret learner interaction. And at the last stage we will collect reasoning in the form of text data. Interviews will also be collected. All this information will help understand the learners learning process and help teachers to develop scaffolds for learning.

2. BACKGROUND

Children come to class already having thought about a variety of events and subjects related to the natural world and try to make sense of their surroundings by constructing mental models. Many researchers use the term "alternative conceptions" since it is value-neutral and express respect for learners’ perspectives [6]. There are also other names that have been suggested, ranging from "naive ideas," "prescientific concepts," "preconceptions," and "conceptual primitives," to the complex "limited or inappropriate propositional hierarchies," or LIPHS [11]. One example of an alternative conception in biology is: Because plants cannot move, young children frequently believe that they are not living, and many older learners believe that life forms like seeds are not living [6]. There are a number of reasons why learners may have these different conceptions, one of which is that they attempt to relate newly learned concepts

M. Pawar and S. Murthy. Understanding learners’ alternative conceptions through interaction patterns during analogical reasoning. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 572–574, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115638

to existing real-world situations which is by using analogies. In order to comprehend how they link two separate situations, it is important to understand learners' reasoning and thought processes.

The task of understanding and addressing learners' alternative conceptions often falls on teachers. Even while some modern teaching-learning approaches place a strong emphasis on self-learning technologies, teachers still play a crucial role in monitoring, scaffolding, and inspiring learners even while using self-learning resources. Being aware of learners' thought processes through their analogical reasoning will help teachers in understanding their alternative conceptions. Eye gaze data and log data of learner engagement with the system will provide nuanced insights like the specific area where the learner has spent more time. This can inform possible learning difficulties that the learner might be facing, like confusion about a concept. This would otherwise be impossible to capture with merely classroom discussion.

3. THEORETICAL FRAMEWORK

A comparison of two objects, or systems of objects, that focuses on the similarities between them is called an analogy [1]. In biology, the analogy is a similarity in function between parts dissimilar in origin and structure. Analogies can be effective teaching aids since they are believed to aid learners in building new knowledge by connecting it to existing knowledge structures [5]. Analogical reasoning is a cognitive process that involves comparing two or more objects to find their commonalities and differences. The activities in the learning environment will be developed on the basis of an analogical reasoning framework known as the Analogical Process Model (APM). Holyoak and Thagard created this framework for using analogies in reasoning. Finding the source analogy, mapping the analogy's structure, and transferring knowledge from the source analogy to the target problem are the three steps of this method [7]. Fig. 1 shows the steps of the framework.

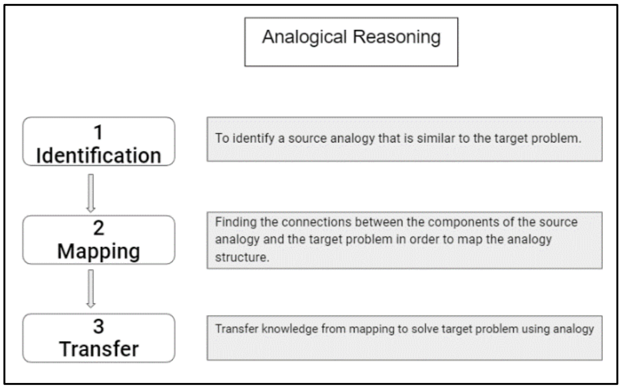

Analogical Reasoning
1 Identification
To identify a source analogy that is similar to the target problem.
2 Mapping
Finding the connections between the components of the source analogy and the target problem in order to map the analogy structure.
3 Transfer
Transfer knowledge from mapping to solve target problem using analogy

Fig. 1 Three steps of analogical reasoning

4. RESEARCH OBJECTIVE

The objective of this study is to understand learners' analogical reasoning through interaction with the proposed learning environment. The primary research questions to investigate are as follows.

RQ1. What do learners' interaction patterns in the Technology Enhanced Learning environment inform us about their analogical reasoning?

RQ2. What are learners' different alternative conceptions as they reason through different analogies?

5. DESIGN OF THE LEARNING ENVIRONMENT

Considering that this study is still in its early developmental phase, the suggested learning environment will be divided into three main stages. The suggested organization of the learning environment is shown in Fig. 2, along with a description of what the teachers and learners would be doing. Stage 1 of the learning environment will be a reading section where learners are supposed to read the content about a particular concept example such as wound healing. The second stage will be designed on the basis of the Analogical Process Model. One scenario for each concept will be designed with activities based on the three steps of APM. The last stage is the reasoning stage where learners will be asked a few questions and they have to write the reasoning behind their actions in stage 2. All the stages will include reflection spots and scaffolding prompts to complete the activates.

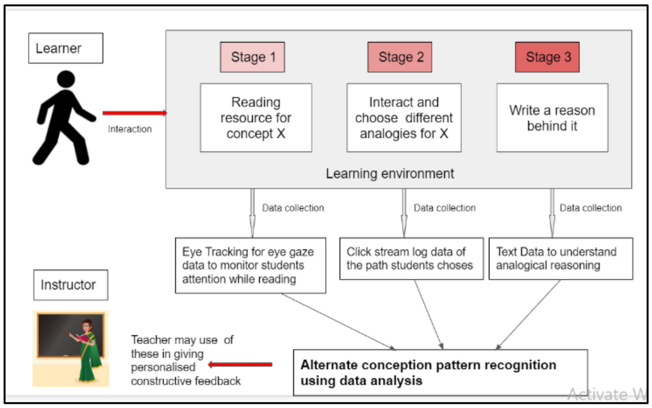

Learner
Interaction
Stage 1
Reading resource for concept X
Stage 2
Interact and choose different analogies for X
Stage 3
Write a reason behind it
Learning environment
Data collection
Data collection
Data collection
Eye Tracking for eye gaze data to monitor students attention while reading
Click stream log data of the path students choses
Text Data to understand analogical reasoning
Instructor
Teacher may use of these in giving personalised constructive feedback
Alternate conception pattern recognition using data analysis
Activate W

Fig. 2 Study design

6. METHODOLOGY

6.1 Target population and sampling

The study participants would be undergraduate bioscience learners who are taking immunology courses. The first concept introduced in this grade is human immunity. About 10 learners will take part in the pilot trial with the learning environment.

6.2 Data Collection and Data Analysis

In this study, a mixed-methods strategy will be applied. Data of two kinds will be gathered. Utilizing click stream data, text-formatted data, and eye gaze tracker, quantitative data will be gathered. The eye-tracking data will reveal which passages in the text the reader spent the most time reading, missed, or skipped. They will engage with the system in a way that is informed by log data. Additionally, textual data can be handled via keyword search. Collectively, these data can be used to comprehend the patterns of various learners. Additionally, after the study, interviews will be conducted to gather additional data that can be used to support interaction patterns.

7. EXPECTED CONTRIBUTIONS

The proposed study will shed insight into individual learners' reasoning. This will inform us about different alternative conceptions of learners in biologically complex systems, such as the human immune system. Alternative conceptions of learners that teachers might have missed or would miss in the classroom can be informed by the study. It will be easy and beneficial for the teacher to provide tailored feedback to one learner or a group of learners and modify their teaching methods once they have learned where and why their learners have alternative conceptions.

8. ASPECTS OF THE RESEARCH ON WHICH ADVICE IS SOUGHT

The suggested research is still in the planning stages. Advice on how to use and analyze the data gathered to determine how various components of learning might be effective.

9. REFERENCES

[1] Bartha, P. (2013). Analogy and analogical reasoning.

[2] Brown, D.E., Clement, J. Overcoming misconceptions via analogical reasoning: abstract transfer versus explanatory model construction. Instr Sci 18, 237–261 (1989). https://doi.org/10.1007/BF00118013

[3] Cheng, C. M., & Chen, C. H. (2009). The analysis of Taiwan assessment of student achievement 2007. *Journal of Educational Research and Development*, *5*(4), 1-38.

[4] Doğru, M. S., & Özsevgeç, L. C. (2018). Biology subjects which the teacher candidates have difficulties in learning and leading reasons. *European Journal of Education Studies*.

[5] Duit, R. (1991). On the role of analogies and metaphors in learning science. *Science education*, *75*(6), 649-672.

[6] Fisher, K. M., & Moody, D. E. (2002). Student misconceptions in biology. In *Mapping biology knowledge* (pp. 55-75). Springer, Dordrecht.

[7] Holyoak, K. J., & Thagard, P. (1997). The analogical mind. *American psychologist*, *52*(1), 35.

[8] James, S., Cogan, P., & McCollum, M. (2019). Team-based learning for immunology courses in allied health programs. *Frontiers in Immunology*, *10*, 2477.

[9] Pradita, D. A. R., Maswar, M., Tohir, M., Junaidi, J., & Hadiyansah, D. N. (2021, February). Analysis of reflective student analogy reasoning in solving geometry problems. In *Journal of Physics: Conference Series* (Vol. 1783, No. 1, p. 012105). IOP Publishing.

[10] Versteeg, M., Wijnen-Meijer, M., & Steendijk, P. (2019). Informing the uninformed: a multitier approach to uncover learners' misconceptions on cardiovascular physiology. *Advances in physiology education*, *43*(1), 7-14.

[11] Wandersee, J. H., Mintzes, J. J., & Novak, J. D. (1994). Research on alternative conceptions in science. *Handbook of research on science teaching and learning*, *177*, 210..

Learning through Wikipedia and Generative AI Technologies

Praveen Garimella
International Institute of Information Technology
Hyderabad, India
praveeng@iiit.ac.in

Vasudeva Varma
International Institute of Information Technology
Hyderabad, India
vv@iiit.ac.in

ABSTRACT

This tutorial will examine the use of Wikipedia and generative AI technologies in asynchronous learning environments. Participants will learn about the research on accountable talk and its impact on student learning, as well as the challenges of implementing the learning principles using Wikipedia in an asynchronous setting. The tutorial will also showcase the potential of generative AI technologies, such as chatbots and language models, to facilitate accountable talk and support student-led discussions in asynchronous learning environments.

By the end of the tutorial, participants will have a solid understanding of the potential of generative AI technologies to enhance student learning and scale accountable talk in asynchronous learning environments. This study conducts a comprehensive analysis of three distinct yet interconnected components that shape contemporary learning environments: Accountable Talk, integration of Wikipedia, and the utilization of generative AI technologies. This investigation aims to highlight the immense potential these elements possess in transforming educational landscapes, particularly within asynchronous learning contexts and in the democratization of knowledge. Additionally, the study explores the societal implications of deploying these methodologies within classrooms, underlining their potential contribution towards the creation of an equitable, knowledgeable, and socially aware society.

Keywords

Asynchronous Learning, Accountable Talk, Wikipedia, LLM

1. INTRODUCTION

In response to the swift transformation of the educational sector, pioneering teaching methodologies that can adapt to various learning environments are of the essence. This study endeavors to offer a comprehensive understanding of three core components: Accountable Talk, the educational role of Wikipedia, and the implementation of generative AI technologies. An exploration of these critical elements facilitates in-depth insights into their influence on learners' communication, social, emotional, and cognitive development, and their capacity to enrich personalized learning experiences.

2. THEORETICAL FOUNDATIONS

This research leans on three primary theoretical foundations: the concept of Accountable Talk, the integration of Wikipedia in academic environments, and the application of generative AI technologies within learning contexts.

Accountable Talk, an instructional approach designed to enhance learning by sparking critical thinking and promoting collaborative discourse, is a cornerstone of the research. [4] This pedagogical methodology centers around students holding themselves responsible for the accurate dissemination of knowledge, sound reasoning, and active community participation, thus boosting their cognitive abilities. The primary objective of Accountable Talk is to refine students' reasoning skills, a competence that is transferable across various academic disciplines. Building upon this conceptual groundwork, the theory of Accountable Talk emphasizes that a community-focused engagement model significantly enhances comprehension and enriches educational outcomes. Implementing this educational strategy requires the creation of a set of ground rules fostering an inclusive learning environment and prompting active intellectual discussions. This context fosters the public exchange of diverse ideas and thoughts, facilitating advanced learning and allowing the identification of misconceptions within the learning community.

2.1 Accountable Talk in Education

1. Maximize Learning Outcomes: Deploying Accountable Talk undoubtedly elevates students' understanding and recall, yielding notable enhancements in their academic performance.

2. Elevate Critical Thinking Skills: Accountable Talk is an exceptional tool for promoting critical thinking. It compels students to delve deeply into subjects, critically dissect assumptions, and construct coherent arguments. Paul and Elder's concept of ”strong sense” critical thinking strongly supports this assertion. [2]

P. Garimella and V. Varma. Learning through wikipedia and generative ai technologies. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 575–577, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115762

3. Promote Active Engagement: By employing Accountable Talk, students are converted from passive absorbers of information into dynamic participants in their learning journey, significantly boosting engagement and comprehension of subjects. This principle aligns with dialogue on the crucial role of classroom dialogue and active participation in improving understanding and engagement. [5]
4. Strengthen Communication Skills: Accountable Talk is instrumental in refining students' communication abilities, empowering them to express their thoughts clearly, listen actively, and respond respectfully to diverse viewpoints. Mercer and Littleton's (2007) viewpoint on dialogue's role in refining reasoning, collaboration, and communication skills further endorses this perspective. [3]
5. Facilitate Collaborative Learning: Accountable Talk offers a well-structured platform for collaborative learning, directing students to collaborate effectively, honor diversity, and expand upon one another's ideas. Johnson and Johnson's (2009) research emphatically backs up the benefits of collaborative learning. [1]
6. Boost Social and Emotional Skills: Accountable Talk plays a crucial role in advancing empathy, respect, and understanding of different viewpoints, hence significantly contributing to the maturation of students' social and emotional capacities.
7. Build a Learning Community: Accountable Talk is a potent tool in fostering an inclusive community of learners. It forms a supportive learning environment that significantly enhances engagement and academic results for all students.
8. Promote Higher-Order Thinking: Accountable Talk is a powerful medium that stimulates students to partake in higher cognitive processes as detailed in Bloom's taxonomy, including analysis, synthesis, and evaluation, instead of mere information memorization. So, make Accountable Talk your standard teaching approach and watch your students thrive!

Integrating Wikipedia assignments into curricula illustrates the practical application of knowledge, thereby refining diverse learner skills. Student contributions to Wikipedia, under instructor guidance, foster research skills and sophisticated understanding of writing for a broad, international readership. The utilization of generative AI technologies, specifically Large Language Models (LLMs), represents a transformative approach to education. With their ability to customize learning experiences and mimic human-like interactions, LLMs hold considerable potential in deepening learning and enhancing teaching methodologies.

3. PRACTICAL IMPLEMENTATION

In leveraging the capabilities of advanced artificial intelligence (AI) systems, this study introduces an innovative pedagogical methodology incorporating Language Learning Models (LLMs), such as GPT-4. The primary purpose is to foster accountable talk and promote information literacy skills among students. Furthermore, this study explores the applicability of such learning models in a practical task: the creation of content for Wikipedia.

The study presents a distinct delineation of roles for both the LLM and the students. The LLM is positioned as a facilitator of knowledge, delivering subject matter expertise, and also as an evaluator of student work. This AI model adheres to protocols of accountable talk, promoting an environment that encourages respectful and evidence-based dialogue. On the other hand, students engage as active participants in the learning process, synthesizing the knowledge provided by the LLM and other reliable sources, and eventually producing content suitable for Wikipedia. A noteworthy aspect of the methodology is the emphasis on the development of information literacy skills.

The pedagogical framework encourages students to conduct their own research, augmenting the knowledge provided by the LLM. In this process, students learn to differentiate between reliable facts and misinformation, and identify potential biases, enhancing their capacity to critically evaluate information. The LLM also plays a crucial role in the content creation process.

Using the information gained through interactions with the LLM and their independent research, students draft Wikipedia articles. The LLM offers support during this process, providing suggestions, refining language and style, and ensuring compliance with Wikipedia's content guidelines. The AI model also evaluates the students' drafts, providing evidence-based feedback in line with accountable talk principles.

This study proposes an educational approach that effectively blends AI technology with pedagogical practices. By integrating LLMs in the learning process, there is potential for enhanced accountable talk and information literacy skills, ultimately fostering an environment conducive to active learning and knowledge synthesis. Future research could further explore the integration of AI in educational settings and evaluate the impacts on student learning outcomes.

4. CONCLUSIONS

By examining Accountable Talk, Wikipedia integration, and the deployment of AI in educational contexts, the study sheds light on potential societal transformations. Implementing these strategies could significantly influence asynchronous learning environments and the democratization of knowledge, ultimately affecting societal outcomes. Promotion of Accountable Talk encourages learners to partake in intellectual dialogues, enhancing critical thinking and collaboration skills. The integration of Wikipedia in educational settings democratizes learning by empowering students worldwide to contribute to a communal knowledge base. Finally, the deployment of generative AI technologies provides individualized learning support, thereby enhancing inclusivity and minimizing educational disparities.

The societal impacts of these strategies reach beyond individual classrooms, signaling towards a future where learning is universally accessible. By optimizing asynchronous learning environments and democratizing education, this study contributes to the cultivation of a society in which equity, digital citizenship, and mutual respect are emphasized.

5. REFERENCES

[1] F. Arató. Towards a complex model of cooperative learning. *Da Investigação às Práticas*, 2013.

[2] D. Hawkins, L. Elder, and R. Paul. *The Thinker's Guide to Clinical Reasoning: Based on Critical Thinking Concepts and Tools*. Rowman & Littlefield, 2019.

[3] N. Mercer and K. Littleton. *Dialogue and the development of children's thinking: A sociocultural approach*. Routledge, 2007.

[4] L. B. Resnick, C. S. Asterhan, and S. N. Clarke. Accountable talk: Instructional dialogue that builds the mind. *Geneva, Switzerland: The International Academy of Education (IAE) and the International Bureau of Education (IBE) of the United Nations Educational, Scientific and Cultural Organization (UNESCO)*, 2018.

[5] G. Wells and R. M. Arauz. Dialogue in the classroom. *The journal of the learning sciences*, 15(3):379–428, 2006.

Introduction to Neural Networks and Uses in EDM

Agathe Merceron
Berlin University of Applied Sciences
merceron@bht–berlin.de

Ange Tato
École de Technologie Supérieure
ange-adrienne.nyamen-tato@etsmtl.ca

ABSTRACT

In this **half-day** tutorial, participants first explore the fundamentals of feed-forward neural networks, such as the backpropagation mechanism; the subsequent introduction to the more complex Long Short Term Memory neural networks builds on this knowledge. The tutorial also covers the basics of the attention mechanism, the Transformer neural networks, and their application in education with Deep Knowledge Tracing. There will be some hands-on applications on open educational datasets. The participants should leave the tutorial with the ability to use neural networks in their research. A laptop capable of installing and running Python and the Keras library is required for full participation in this half-day tutorial.

Keywords

Neurons, Neural networks, LSTM, Attention mechanism, Transformers

1. INTRODUCTION

Neural networks (NN) are as old as the relatively young history of computer science: McCullogh and Pitts already proposed nets of abstract neurons in 1943 as Haigh and Priestley report in [7]. However, their successful use, especially in the form of convolutional neural networks (CNN), Long Short Term Memory (LSTM), or Transformer neural networks, in areas such as image recognition, language translation, or chatbot in the last years has made them widely known, also in the Educational Data Mining (EDM) community. This is reflected in the contributions that are published each year in the proceedings of the conference.

In [11], we counted the percentage of the contributions in the EDM proceedings of the Educational Data Mining (EDM) conference from the beginning of the conference in 2008 till 2019 (long and short papers, posters and demos, young research track, doctoral consortium, and papers of the industry track) that have used some kind of neural networks in their research. While the percentage stayed below 10% till 2015, it started to increase in 2016 to reach 28% in 2019. This trend has continued since then with 14 long papers from 26 mentioning some kind of neural networks in their research in the EDM proceedings of 2022.

Recognizing the growing importance of neural networks in the EDM community, this tutorial aims to provide 1) an introduction to neural networks in general and to LSTM neural networks with a focus on the attention mechanism and the Transformer neural networks and 2) a discussion venue on these exciting techniques. Compared with our precedent tutorial [11], the main difference is the introduction to Transformer neural networks. This tutorial targets 1) participants who have no or very little prior knowledge about neural networks and would like to use them in their future work or would like to better understand the work of others, and 2) participants interested in exchanging and discussing their experience with the use of neural networks. A simple kind of neural network is a feedforward neural network also often called a multilayer perceptron. It propagates the calculation of each neuron from its inputs through all layers in a directed way forward to its outputs. In education, such a NN has been used, for example, to predict the performance of students. The work of Romero et al. [18] presented at the first EDM conference in 2008 uses it to predict the final mark of students in a course taught with the support of the learning platform Moodle. while the work of Wagner et al. [24] uses it to predict whether students will drop out of a study program.

While their primary use was in Natural Language Processing (NLP) Tasks, LSTM neural networks have been extensively used in education and have achieved remarkable results [22, 20, 6]. Unlike feedforward neural networks that cannot remember the past, LSTM have cycles and are recurrent neural networks. The LSTM [9] architecture can learn long-term dependencies using a memory cell that can preserve states over long periods. It is suitable for contexts where sequential information and temporal prediction is important such as in education, where we are interested in predicting students' outcome based on past behavior. Deep Knowledge Tracing [14] is probably the best example of using LSTM to track a student's state of knowledge while interacting with a tutoring system. Numerous variants of LSTM have been proposed, such as the Gated Recurrent Unit (GRU) [4] or the LSTM combined with the attention mechanism, especially the Transformer neural networks [23].

A. Merceron and A. Tato. Introduction to neural networks and uses in edm. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 578–581, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115774

Attention [3] in machine learning refers to a model's ability to focus on specific elements in data. It helps the LSTM to learn where to look in the data. It was initially designed in Neural Machine Translation using sequence-to-sequence (Seq2Seq or encoder-decoder) [19] models. However, since the attention mechanism can improve the prediction results of NN models, it is now widely used in text mining in general. Especially in the education domain, it has been used for question-answering tasks, sequential modeling for student performance prediction, or to predict essay or short answer scoring [25, 17]. Transformer neural networks aim to solve sequence-to-sequence tasks while handling long-range dependencies. It uses the attention mechanism and GPU (Graphics Processing Unit) computing. The input sequence of the Transformer neural network can be passed parallelly, which speeds up the training. It can also overcome the vanishing gradient issue thanks to its multi-headed attention layer. The use of transformers in education is only in its infancy. However, given its notable results (e.g., Generative Pre-trained Transformer (GPT)[2], Bidirectional Encoder Representations from Transformers (BERT)[10]), we think that we will see an increasing number of research papers using this architecture in EDM.

2. PROPOSED FORMAT

Table 1: Timeline and activities

Time	Item
45 minutes	**Presentation**: introduction - Feedforward neural networks and backpropagation
45 minutes	Application - Discussion - Hands-on
30 minutes	Break
60 minutes	**Presentation**: LSTM, Attention Mechanism, and Transformer
60 minutes	Application - Implementation of a LSTM for student performance prediction - Discussion

3. DESCRIPTION OF THE TUTORIAL

3.1 Introduction to feed-forward neural networks

This part begins with artificial neurons and their structure - inputs, weight, output, and activation function - and the calculations that are feasible and not feasible with one neuron only. It continues with feedforward neural networks or multi-layer perceptrons (MLP). A hands-on example taken from [8] illustrates how a feedforward neural network calculates its output. Further, this part introduces loss functions and the backpropagation algorithms and makes clear what a feedforward neural network learns. Backpropagation is demonstrated with the hands-on example introduced before.

3.2 Application of feedforward NN

This part discusses the use of feedforward neural networks in EDM research. These networks are often used to predict students' performance and students at risk of dropping out, see for example [5, 1, 24]. It must be noted that feedforward neural networks do not necessarily give better results than other algorithms for this kind of task. Other uses emerge. For example, Ren et al. use them to model the influence on the grade of a course taken by a student on all other courses that the student has co-taken [16]. As another example, Or and Russel [13] uses intentionally a feedforward "neural network model to both automatically assess the design of a program and provide personalized feedback to guide students on how to make corrections".

It must be noted that neural networks are considered not interpretable, see [12]. When explanations are crucial, it might be worthwhile to evaluate whether interpretable algorithms might be used instead; another way is to generate explanations with other algorithms, see [20] for challenges in doing so.

The main activity of this part is for participants to solve a classification task on an educational dataset; participants will create, inspect and evaluate a feedforward neural network with Python and relevant libraries.

3.3 LSTM

In this part of the tutorial, basic concepts of LSTM are covered. We will focus on how the architecture of different elements (cell, state, etc.) works. Participants will learn how to use an LSTM for the prediction of learners' outcomes in an educational system. Concepts such as Deep Knowledge Tracing (DKT) will also be covered.

3.4 Attention Mechanism

In this part, the attention mechanism is introduced. Participants will learn how this mechanism works and how to use it in different cases. We will explore concepts such as global and local attention in neural networks.

3.5 Transformer neural networks

This part introduces the Transformer neural network architecture. Concepts such as multi-headed attention layer and parallel inputs with the use of GPU will be covered.

3.6 Application

This hands-on part will explore existing real-life applications of LSTM (especially Deep Knowledge Tracing and Knowledge tracing with transformer) in education. We will also explore the combination of LSTM with Expert Knowledge (using the attention mechanism) for Predicting Socio-Moral Reasoning skills [21, 22]. Participants will implement an LSTM, especially a Transformer, with an attention mechanism for the prediction of students' performance in a tutoring system [15]. We will use Keras (Python) library for coding and also use open educational datasets (e.g., Assistments benchmark dataset).

3.7 Objectives and outcomes

The objectives of this tutorial are twofold: 1) introduce the fundamental concepts and algorithms of neural networks to newcomers and then build on these fundamentals to give them some understanding of LSTM and the attention mechanism, especially the Transformer neural networks; 2) provide a place to discuss and exchange about experiences while using neural networks with educational data. Newcomers should leave the tutorial with a good understanding of neural networks and the ability to use them in their own research or to appreciate better research works that use neural networks. Participants already knowledgeable about neural networks get a chance to discuss and share about this topic and connect with others. A website will be created to display important information to participants: schedule, slides, data, and software to download and install.

4. SHORT BIOGRAPHIES

Agathe Merceron is a Professor of Computer Science at Berlin University of Applied Sciences teaching courses such as machine learning. She was head of the online study program "Computer Science and Media" (Bachelor and Master) till March 31, 2022. Her research interest is in Technology Enhanced Learning with a focus on Educational Data Mining and Learning Analytics. She has served as a program chair for national and international conferences and workshops, in particular for the international conferences Educational Data Mining and Learning Analytics and Knowledge. She is Editor of the Journal of Educational Data Mining and member of the board of the Journal "Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation" (STICEF).

Ange Tato is a Senior Lecturer in computer science at École de Technologie Supérieure de Montréal. She has worked as a research scientist in machine learning at Bem Me Up Augmented Intelligence Montreal for 4 years. Her research interest is in the fundamentals of machine learning algorithms applied to user modeling in intelligent systems. Some of her notable works focus on improving first-order optimization algorithms (with gradient descent); improving neural network architectures for multimodal data to predict or classify user behaviors (players, learners, etc.) in adaptive intelligent systems; and integrating expert knowledge into deep learning models to improve their predictive power and for better traceability of these models. She has served as Poster and Demo Track Co-Chair for Educational Data Mining 2021, Program Committee Member of international conferences such as ICCE, or AIED.

5. REFERENCES

[1] J. Berens, K. Schneider, S. Gortz, S. Oster, and J. Burghoff. Early detection of students at risk - predicting student dropouts using administrative student data from german universities and machine learning methods. *Journal of Educational Data Mining*, 11(3):1–41, 12 2019.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. Attention-based models for speech recognition. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 577–585, Cambridge, MA, USA, 2015. MIT Press.

[4] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*. MIT Press, 2014.

[5] G. Dekker, M. M. Pechenizkiy, and J. Vleeshouwers. Predicting students drop out: A case study. In T. Barnes, M. Desmarais, C. Romero, and S. Ventura, editors, *Proceedings of the second International Conference on Educational Data Mining (EDM 2009)*, pages 41–50. International Educational Data Mining Society, July 2009.

[6] A. Ghosh, N. Heffernan, and A. S. Lan. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2330–2339, 2020.

[7] T. Haigh and M. Priestley. von neumann thought turing's universal machine was' simple and neat.' but that didn't tell him how to design a computer. *Communications of the ACM*, 63(1):26–32, 2019.

[8] J. Han, M. Kamber, and J. Pei. *Data Mining - Concepts and Techniques*. Morgan Kaufmann, 2012.

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[10] J. D. M.-W. C. Kenton and L. K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, pages 4171–4186, 2019.

[11] A. Merceron and A. Tato. An introduction to neural networks. In A. N. Rafferty, J. Whitehill, C. Romero, and V. Cavalli-Sforza, editors, *Proceedings of the International Conference on Educational Data Mining (EDM 2020)*, pages 821–823. International Data Mining Society, 2020.

[12] C. Molnar. Interpretable machine learning. https://christophm.github.io/interpretable-ml-book/, Nov 2022. Last checked on Dec 07, 2022.

[13] J. W. Orr and N. Russell. Automatic assessment of the design quality of python programs with personalized feedback. In I.-H. S. Hsiao, S. S. Sahebi, F. B. chet, and J.-J. Vie, editors, *Proceedings of the 14th International Conference on Educational Data Mining (EDM 2021)*, pages 495–501. International Educational Data Mining Society, July 2021.

[14] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 505–513, Cambridge, MA, USA, 2015. MIT Press.

[15] S. Pu, M. Yudelson, L. Ou, and Y. Huang. Deep knowledge tracing with transformers. In *International Conference on Artificial Intelligence in Education*, pages 252–256. Springer, 2020.

[16] Z. Ren, X. Ning, A. Lan, and H. Rangwala. Grade prediction based on cumulative knowledge and co-taken courses. In M. Desmarais, C. F. Lynch, A. Merceron, and R. Nkambou, editors, *Proceedings of the 12th International Conference on Educational Data Mining (EDM 2019)*, pages 158–167. International Educational Data Mining Society, July 2019.

[17] B. Riordan, A. Horbach, A. Cahill, T. Zesch, and C. Lee. Investigating neural architectures for short answer scoring. In *Proceedings of the 12th workshop on innovative use of NLP for building educational applications*, pages 159–168, 2017.

[18] C. Romero, S. Ventura, P. Espejo, and C. Hervás.

Data mining algorithms to classify students. In R. S. J. de Baker, T. Barnes, and J. E. Beck, editors, *Proceedings of the first International Conference on Educational Data Mining (EDM 2008)*, pages 8–17. International Data Mining Society, 2008.

[19] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

[20] V. Swamy, B. Radmehr, N. Krco, M. Marras, and T. Käser. Evaluating the explainers: Black-box explainable machine learning for student success prediction in MOOCs. In N. Bosch and A. Mitrovic, editors, *Proceedings of the 15th International Conference on Educational Data Mining (EDM 2022)*, pages 98–109, Durham, United Kingdom, July 2022. International Educational Data Mining Society.

[21] A. Tato and R. Nkambou. Infusing expert knowledge into a deep neural network using attention mechanism for personalized learning environments. *Frontiers in Artificial Intelligence*, 5:921476, 2022.

[22] A. A. N. Tato, R. Nkambou, and A. Dufresne. Hybrid deep neural networks to predict socio-moral reasoning skills. In M. Desmarais, C. F. Lynch, A. Merceron, and R. Nkambou, editors, *Proceedings of the 12th International Conference on Educational Data Mining (EDM 2019)*, pages 623–626. International Educational Data Mining Society, 2019.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[24] K. Wagner, A. Merceron, and P. Sauer. Accuracy of a cross-program model for dropout prediction in higher education. In *Companion Proceedings of the 10th International Learning Analytics & Knowledge Conference (LAK 2020)*, pages 744–749, 2020.

[25] X. Xiong, S. Zhao, E. G. Van Inwegen, and J. E. Beck. Going deeper with deep knowledge tracing. In T. Barnes, M. Chi, and M. Feng, editors, *Proceedings of the International Conference on Educational Data Mining (EDM 2016)*, pages 545–550. International Data Mining Society, 2016.

How to Open Science: Promoting Principles and Reproducibility Practices within the Educational Data Mining Community

Aaron Haim
Worcester Polytechnic Institute
100 Institute Road
Worcester, Massachusetts, USA
ahaim@wpi.edu

Stacy T. Shaw
Worcester Polytechnic Institute
100 Institute Road
Worcester, Massachusetts, USA
sshaw@wpi.edu

Neil T. Heffernan
Worcester Polytechnic Institute
100 Institute Road
Worcester, Massachusetts, USA
nth@wpi.edu

ABSTRACT

Across the past decade, open science has increased in momentum, making research more openly available and reproducible. Educational data mining, as a subfield of education technology, has been expanding in scope as well, developing and providing better understanding of large amount of data within education. However, open science and educational data mining do not often intersect, causing a bit of difficulty when trying to reuse methodologies, datasets, analyses for replication, reproduction, or an entirely separate end goal. In this tutorial, we will provide an overview of open science principles and their benefits and mitigation within research. In the second part of this tutorial, we will provide an example on using the Open Science Framework to make, collaborate, and share projects. The final part of this tutorial will go over some mitigation strategies when releasing datasets and materials such that other researchers may easily reproduce them. Participants in this tutorial will gain a better understanding of open science, how it is used, and how to apply it themselves.

Keywords

Open Science, Reproducibility, Preregistration

1. BACKGROUND

Open Science is a term used to encompass making methodologies, datasets, analyses, and results of research publicly accessible for anyone to use freely[6, 14]. This term started to frequently occur in the early 2010s when researchers began noticing that they were unable to replicate or reproduce prior work done within a discipline[13]. There also tended to be a large amount of ambiguity when trying to understand what process was followed to conduct a study or whether a specific material was used but not clearly defined. Open science, as a result, started to gain more traction to provide greater context, robustness, and reproducibility metrics with each subtopic encompassed under the term receiving their own formal definition and usage. The widespread adoption of open science began to explode exponentially when large scale studies conducted in the mid 2010s found that numerous works were difficult or impossible to reproduce and replicate in psychology[2] and other disciplines[1].

Some principles commonly referred to as part of open science and its processes: open data, open materials, open methodology, and preregistration. **Open Data** specifically targets datasets and their documentation for public use without restriction, typically under a permissive license or in the public domain[8]. Not all data can be openly released (such as with personally identifiable information); but there are specifications for protected access that allow anonymized datasets to be released or a method to obtain the raw dataset itself. **Open Materials** is similar in regard except for targeting tools, source code, and their documentation[5]. This tends to be synonymous with **Open Source** in the context of software development, but materials are used to encompass the source in addition to available, free-to-use technologies. **Open Methodology** defines the full workflow and processes used to conduct the research, including how the participants were gathered, what was told to them, how the collected data was analyzed, and what the final results were[6]. The methodologies typically expand upon the original paper, such as technicalities that would not fit in the paper format. Finally, **Preregistration** acts as an initial methodology before the start of an experiment, defining the process of research without knowledge of the outcomes[10, 11]. Preregistrations can additionally be updated or created anew to preserve the initial experiment conducted and the development as more context is generated.

2. TUTORIAL GOALS

Open science principles and reproducibility metrics are becoming more commonplace within numerous scientific disciplines. Within many subfields of educational technology, such as educational data mining, however, the adoption and review of these principles and metrics are neglected or sparsely considered[9]. There are some subfields of education technology that have taken the initiative to introduce open science principles (special education[3]; gamification[4], education research[7]); however, other subfields have seen little to no adoption. Concerns and inexperience in what can be made

A. Haim, S. Shaw, and N. Heffernan. How to open science: Promoting principles and reproducibility practices within the educational data mining community. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 582–584, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115776

publicly available to how to reproduce another's work are some of the few reasons why researchers may choose to avoid or postpone discussion on open science and reproducibility. On the other hand, lack of discussion can lead to tediousness and repetitive communication for datasets and materials or cause a reproducibility crisis[1] within the field of study. As such, there is a need for accessible resources and understanding on open science, how it can be used, and how to mitigate any potential issues that may arise within one's work at a later date.

Admitting our own initial lack of proper adoption and reproducibility first, in this tutorial, we will cover some of the basic principles of open science and some of the challenges and mitigation strategies associated with education technology specifically. Next, we will provide a step-by-step explanation on using the Open Science Framework to create a project, collaborate with other researchers, post content, and preregister a study. Using examples from the field of educational technology, we will showcase how to incorporate open science principles, in addition to practices that, when implemented, would improve reproducibility.

This tutorial will build and expand on a prior, successful tutorial at the *15th International Conference on Educational Data Mining* in 2022[1][12] and an accepted tutorial to be presented at the *13th International Conference on Learning Analytics and Knowledge* in 2023[2].

3. TUTORIAL ORGANIZATION

The tutorial will occur over half a day and focuses on introducing some common open science principles and their usage within education technology, providing an example on using the Open Science Framework to create a project, post content, and preregister studies, and using previous papers to apply the learned principles and any additional reproduction mitigation strategies. An outline of this tutorial can be found below:

- First, we will provide a presentation on an overview of a few problems when conducting research. Using this as a baseline, we will introduce open science and its principles and how they can be used to nullify some of these issues and mitigate others. In addition, we will attempt to dispel some of the misconceptions of these principles.

- Second, we will provide a live example of using the Open Science Framework (OSF) website to make an account, create a project, add contributors, add content and licensing, and publicize the project for all to see. Afterwards, we will provide a guide to creating a preregistration, explaining best practices, and identifying how to create an embargo. Additional features and concerns, such as anonymizing projects for review and steps required to properly do so, will be shown.

- Third, we will discuss reproducibility metrics within work when providing datasets and materials. This will review commonly used software and languages (e.g. Python, RStudio) and how, without any steps taken, most work tends to be extremely tedious to reproduce or are not reproducible in general. Afterwards, we will provide some mitigation strategies needed to remove these concerns.

- Finally, we will take some existing papers either from the author's own research or from prior education technology conferences that do not meet some open science principles or cannot easily be reproduced and apply what has been learned across the entire tutorial. We will use a few papers, each containing different issues, and apply the necessary steps needed to reproduce the results within the paper.

3.1 Dissemination of Information

The dissemination of information for this tutorial will be provided before and after the conference. Before the conference, information about the tutorial itself will be stored on an OSF project, containing references to the papers used within the final part of the tutorial, any slides to be used within the conference, and additional resources that could provide better understanding of the issues and nuances of avoiding open science and reproducibility metrics. A website separate to the OSF project will also be set up containing the following information for ease of consumption; however, this will only be used as an alternative to the project in case the website disappears at some point in the future.

After the conference, any resources created or recordings taken will be uploaded to the project for preservation. Alternative links will be provided to separate sites for more formal hosting (e.g. videos on YouTube). As this tutorial wants to repeat and expand upon open science and reproducibility at prior workshops across conferences, an additional project will be created on the OSF website containing components pointing to all previous conferences and resources discussed.

3.2 Organizers

Aaron Haim[3] is a Ph.D. student in Computer Science at Worcester Polytechnic Institute. His initial research focuses on developing software and running experiments on crowdsourced, on-demand assistance in the form of hints and explanations. His secondary research includes reviewing, surveying, and compiling information related to open science and reproducibility across papers published at education technology and learning science conferences.

Stacy T. Shaw[4] is an Assistant Professor of Psychology and Learning Sciences at Worcester Polytechnic Institute. She is an ambassador for the Center for Open Science, a catalyst for the Berkeley Initiative in Transparency in Social Sciences, and serves on the EdArXiv Preprint steering committee. Her research focuses on mathematics education, student experiences, creativity, and rest.

Neil T. Heffernan[5] is the William Smith Dean's Professor of Computer Science and Director of the Learning Sciences & Technology Program at Worcester Polytechnic Institute.

[1] https://osf.io/m7cnr/

[2] https://doi.org/10.17605/osf.io/kyxba

[3] https://ahaim.ashwork.net/

[4] http://stacytshaw.com/

[5] https://www.neilheffernan.net/

He is the founder of ASSISTments, an online learning platform which provides immediate feedback for students along with actionable data for teachers. Heffernan has been pushing open science with his graduate students in recent years. He has also started to push the Educational Data Mining committee to broaden their promotion and support of open science.

4. REFERENCES

[1] M. Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454, May 2016.

[2] O. S. Collaboration. Estimating the reproducibility of psychological science. *Science*, 349(6251):aac4716, 2015.

[3] B. G. Cook, L. W. Collins, S. C. Cook, and L. Cook. A replication by any other name: A systematic review of replicative intervention studies. *Remedial and Special Education*, 37(4):223–234, 2016.

[4] A. García-Holgado, F. J. García-Peñalvo, C. de la Higuera, A. Teixeira, U.-D. Ehlers, J. Bruton, F. Nascimbeni, N. Padilla Zea, and D. Burgos. Promoting open education through gamification in higher education: The opengame project. In *Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality*, TEEM'20, page 399–404, New York, NY, USA, 2021. Association for Computing Machinery.

[5] J. Johnson-Eilola. Open source basics: Definitions, models, and questions. In *Proceedings of the 20th Annual International Conference on Computer Documentation*, SIGDOC '02, page 79–83, New York, NY, USA, 2002. Association for Computing Machinery.

[6] P. Kraker, D. Leony, W. Reinhardt, and G. Beham. The case for an open science in technology enhanced learning. *International Journal of Technology Enhanced Learning*, 3(6):643–654, 2011.

[7] M. C. Makel, K. N. Smith, M. T. McBee, S. J. Peters, and E. M. Miller. A path to greater credibility: Large-scale collaborative education research. *AERA Open*, 5(4):2332858419891963, 2019.

[8] P. Murray-Rust. Open data in science. *Nature Precedings*, 1(1):1, Jan 2008.

[9] B. Nosek. Making the most of the unconference, 2022.

[10] B. A. Nosek, E. D. Beck, L. Campbell, J. K. Flake, T. E. Hardwicke, D. T. Mellor, A. E. van 't Veer, and S. Vazire. Preregistration is hard, and worthwhile. *Trends in Cognitive Sciences*, 23(10):815–818, Oct 2019.

[11] B. A. Nosek, C. R. Ebersole, A. C. DeHaven, and D. T. Mellor. The preregistration revolution. *Proceedings of the National Academy of Sciences*, 115(11):2600–2606, 2018.

[12] S. Shaw and A. Sales. Using the open science framework to promote open science in education research. In *Proceedings of the 15th International Conference on Educational Data Mining*, page 853–853. International Educational Data Mining Society, Jul 2022.

[13] B. A. Spellman. A short (personal) future history of revolution 2.0. *Perspectives on Psychological Science*, 10(6):886–899, 2015. PMID: 26581743.

[14] R. Vicente-Saez and C. Martinez-Fuentes. Open science now: A systematic literature review for an integrated definition. *Journal of Business Research*, 88:428–436, 2018.

Data Efficient Machine Learning for Educational Content Creation

Ganesh Ramakrishnan
IIT Bombay
ganesh@cse.iitb.ac.in

Ayush Maheshwari
IIT Bombay
ayusham@cse.iitb.ac.in

ABSTRACT

Machine Learning has revolutionized education by offering numerous practical applications. One such application is Neural Machine Translation (NMT) systems in the field of education, which hold immense social importance. These systems have the potential to make information accessible to diverse users in multilingual societies. By effectively translating audio, video, and textual content into vernacular languages, NMT systems greatly assist both students and teachers. However, when it comes to translating higher education or technical textbooks and courses, it becomes crucial for MT systems to adhere closely to the specific lexicon of the source and target domains. In this tutorial, we present our approach and framework to enable domain-aware translation without the need for parallel domain corpus. We will demonstrate several use-cases and applications that is widely used by hundreds of translators. We will also present our post-editing tool that assists translators in quickly correcting the machine translated text and reduce the cognitive load of users.

Keywords

neural machine translation, OCR, dictionary generation, human-in-the-loop learning, data efficient machine learning

1. INTRODUCTION

The field of Neural Machine Translation (NMT) has achieved remarkable success in achieving state-of-the-art translation capabilities across various language pairs [1]. However, in domain-specific scenarios such as technical content translation, the generic NMT pipeline falls short in guaranteeing the inclusion of specific terms in the translation output. Inclusion of a pre-specified vocabulary becomes crucial for ensuring practical and reliable machine translation (MT). While incorporating domain-specific terms has been relatively easier in phrase-based statistical MT, it poses a challenge in NMT due to the complexity of directly manipulating output representations from the decoder [8]. As an alternative, domain-specific NMT systems have been proposed to generate translations that are aware of the domain by fine-tuning generic NMT models using domain-specific parallel text. However, this approach requires curating translation pairs for each domain, which demands significant human effort and increases the cost of maintaining separate models for each domain. Therefore, it is essential for the MT output to adhere to the source domain by adopting domain-specific terminology, thus reducing and potentially guiding the post-editing effort in translation.

To address this issue, lexically constrained techniques have been employed in NMT, incorporating pre-specified words and phrases in the translation output [4, 3, 2]. In addition to the source sentence, word or phrasal constraints in the target language are provided as input. These constraints can be derived from in-domain source-target dictionaries or can be user-provided source-target constraints during interactive machine translation. Often, these constraints may encode multiple potential translations for a given source phrase. For example, the word 'speed' can be translated into 5 different Hindi phrases *teja, dauḍa, gati, raphtār, cāla* in the physics domain. However, existing constrained translation approaches do not accommodate such ambiguity in the constraints.

2. IMPACT OF THE WORK

The project `https://udaanproject.org` is an end-to-end Machine Translation Framework that includes extensive use of OCR, lexical resources, data efficient learning (open sourced at `https://decile.org`) and a human-in-the-loop machine learning based post-editing platform. This project is an outcome of our Data Efficient Machine Learning[5, 6] (`https://decile.org`) and Natural Language Processing from our group at IIT Bombay. The Udaan project is being used extensively by several, including AICTE (`https://www.aicte-india.org`) for speedy translation of 100s of textbooks into multiple Indian languages. MoUs are also being signed with several state governments - Govt of Maharashtra entered into agreement for usage of `https://udaanproject.org` in the presence of Governor, Education Minister and Director IITB (see `https://udaanproject.org/MediaCoverage?type=mou`)

In this tutorial, we provide insights from our translation ecosystem (https://udaanproject.org) that has helped in translating 100s of diploma and engineering books each in more than 11 Indian languages. We will provide the audience with

G. Ramakrishnan and A. Maheshwari. Data efficient machine learning for educational content creation. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 585–587, Bengaluru, India, July 2023. International Educational Data Mining Society.

`https://doi.org/10.5281/zenodo.8115780`

the holistic view of:

1. How to build a domain-specific lexicon in 11 Indian languages using a small seed dictionary by utilising the innate connection across languages
2. How to build an multilingual NMT model that ingests domain-specific lexicon without affecting the fluency of the predicted sentence
3. How to build a human-in-the-loop AI post-editing tool that benefits from complex OCR (Optical character recognition) and layout analysis to preserve bounding boxes in the source document. and that learns from the user edits and calibrates the output for subsequent occurrences.
4. What are the insights gathered from translating sample 50 books across 11 languages?

The ecosystem at `https://udaanproject.org` that will be presented as a tutorial is fueled by several peer reviewed publications (`https://udaanproject.org/Publications`).

3. UDAAN POST EDITING TOOL

We present UDAAN, an open-source post-editing tool designed to streamline the manual editing process and facilitate the production of high-quality documents in multiple Indic languages[7]. Post-editing lengthy documents that have been translated is often a laborious task, as editors face difficulties in maintaining consistency between the translated and source texts within the document. Existing tools, although retaining source-target user edits through translation memory (TM), fail to provide consistent suggestions throughout the document.

UDAAN offers an end-to-end Machine Translation (MT) plus post-editing pipeline, allowing users to upload a document and obtain raw MT output. Subsequently, users can utilize our tool to edit the raw translations. UDAAN incorporates several advantageous features:

1. Domain-aware, vocabulary-based lexical constrained MT.
2. Source-target and target-target lexicon suggestions for users, employing lexicon alignment between the source and target texts for replacements.
3. Translation suggestions based on user interaction logs.
4. Source-target sentence alignment visualization, reducing cognitive load during the editing process.
5. Translated outputs available in multiple formats, including docs, LaTeX, and PDF.

Our tool offers several advantages: Firstly, it generates domain-aware raw MT by applying lexical constraints to the translations using domain-specific vocabulary. Secondly, users can incorporate lexicons from both the source-target language and the target-target language. Lexicon-based replacements are determined through alignment between the source and target texts. Additionally, the tool continuously records target-target edits made by users, which can be utilized as suggestions within the tool. Thirdly, the tool leverages user edits to improve translation suggestions. Fourthly, the rich text editor of UDAAN includes sentence alignment visualization between the source and target texts, simplifying the editing process and reducing cognitive load. Lastly, users can download the output document in various formats, including docx, LaTeX, and PDF.

Furthermore, UDAAN provides access to approximately 100 in-domain dictionaries to facilitate lexicon-aware machine translation. Although our experiments are limited to English-to-Hindi translation, the tool is language-agnostic. Based on user feedback and experimental results, UDAAN has demonstrated a significant reduction in translation time, approximately three times faster than the baseline method of translating documents from scratch. UDAAN is available for both Windows and Linux platforms, with its source code accessible on our website at Our tool is available for both Windows and Linux platforms. The tool is open-source under MIT license, and the source code can be accessed from our website, `https://www.udaanproject.org`. Demonstration and tutorial videos for various features of our tool can be accessed here. Our MT pipeline can be accessed at `https://udaaniitb.aicte-india.org/udaan/translate/`.

3.1 Acknowledgments

Ayush Maheshwari is supported by a Fellowship from Ekal Foundation (`www.ekal.org`). Ganesh Ramakrishnan is grateful to NLTM OCR Bhashini project as well as the IIT Bombay Institute Chair Professorship for their support and sponsorship.

4. REFERENCES

[1] L. Barrault, O. Bojar, M. R. Costa-jussà, C. Federmann, M. Fishel, Y. Graham, B. Haddow, and M. a. H. Huck. Findings of the 2019 conference on machine translation (WMT19). page 9. Frontiers, 2018.

[2] G. Chen, Y. Chen, Y. Wang, and V. O. Li. Lexical-constraint-aware neural machine translation via data augmentation. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3587–3593, 2021.

[3] G. Dinu, P. Mathur, M. Federico, and Y. Al-Onaizan. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, 2019.

[4] C. Hokamp and Q. Liu. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, 2017.

[5] A. Maheshwari, O. Chatterjee, K. Killamsetty, G. Ramakrishnan, and R. Iyer. Semi-supervised data programming with subset selection. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4640–4651, 2021.

[6] A. Maheshwari, K. Killamsetty, G. Ramakrishnan, R. Iyer, M. Danilevsky, and L. Popa. Learning to

robustly aggregate labeling functions for semi-supervised data programming. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1188–1202, 2022.

[7] A. Maheshwari, A. Ravindran, V. Subramanian, and G. Ramakrishnan. Udaan-machine learning based post-editing tool for document translation. In *Proceedings of the 6th Joint International Conference on Data Science & Management of Data (10th ACM IKDD CODS and 28th COMAD)*, pages 263–267, 2023.

[8] R. H. Susanto, S. Chollampatt, and L. Tan. Lexically constrained neural machine translation with levenshtein transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3536–3543, 2020.