Predicting semantic changes in abstraction in tutor responses to students

Michael Lipschultz*

University of Pittsburgh,
6311 Sennott Square, 210 South Bouquet Street,
Pittsburgh, PA 15260, USA
Email: mil28@pitt.edu
*Corresponding author

Diane Litman

741 Learning Research & Development Center
University of Pittsburgh,
3939 O'Hara Street,
Pittsburgh, PA 15260, USA
Email: litman@cs.pitt.edu

Sandra Katz

733 Learning Research & Development Center,
University of Pittsburgh,
3939 O'Hara Street,
Pittsburgh, PA 15260, USA
Email: katz@pitt.edu

Patricia Albacete

708 Learning Research & Development Center,
University of Pittsburgh,
3939 O'Hara Street,
Pittsburgh, PA 15260, USA
Email: palbacet@pitt.edu

Pamela Jordan

701 Learning Research & Development Center,
University of Pittsburgh,
3939 O'Hara Street,
Pittsburgh, PA 15260, USA
Email: pjordan@pitt.edu

Abstract: Post-problem reflective tutorial dialogues between human tutors and students are examined to predict when the tutor changed the level of abstraction from the student's preceding turn (i.e., used more general terms or more specific terms); such changes correlate with learning. Prior work examined lexical changes in abstraction. In this work, we consider semantic changes. Since we are interested in developing a fully-automatic computer-based tutor, we use only automatically-extractable features (e.g., percent of domain words in student turn) or features available in a tutoring system (e.g., correctness). We find patterns that predict tutor changes in abstraction better than a majority class baseline. Generalisation is best-predicted using student and reflection features. Specification is best-predicted using student and problem features.

Keywords: intelligent tutoring systems; ITS; natural language processing; NLP; abstraction changes; reflective tutorial dialogues; semantic changes; learning technologies.

Reference to this paper should be made as follows: Lipschultz, M., Litman, D., Katz, S., Albacete, P. and Jordan, P. (2014) 'Predicting semantic changes in abstraction in tutor responses to students', *Int. J. Learning Technology*, Vol. 9, No. 3, pp.281–303.

Biographical notes: Michael Lipschultz is a graduate student in the Computer Science Department at the University of Pittsburgh. His current research focuses on enhancing intelligent tutoring systems through student modelling and adaptation, multiple graphical representations, and machine learning.

Diane Litman is Professor of Computer Science, Senior Scientist with the Learning Research and Development Center, and faculty with the Graduate Program in Intelligent Systems, all at the University of Pittsburgh. Her current research focuses on enhancing the effectiveness of educational technology through the use of spoken and natural language processing, affective computing, and machine learning and other statistical methods. Her current projects include AI-enhanced peer review, spoken tutorial dialogue systems, and automated essay assessment.

Sandra Katz received her Doctor of Arts degree in English from Carnegie Mellon University, MS in Information Science from the University of Pittsburgh and an MA in Anthropology, also from the University of Pittsburgh. She is a Research Associate at the University of Pittsburgh's Learning Research and Development Center, where she has directed the development and evaluation of intelligent tutoring systems to support technical training and instruction in various domains. Her research areas include developing adaptive tutorial dialogue systems, increasing the participation of women and minorities in the sciences, and training for careers in technology and medicine.

Patricia Albacete is a Research Associate at the University of Pittsburgh Learning Research and Development Center. Her research has focused on the development and evaluation of intelligent tutoring systems particularly in the area of conceptual physics instruction. She is currently involved in several projects where she is exploring the particular features of human tutoring dialogues that make them effective with the goal of emulating them in computer tutors.

Pamela Jordan received her PhD in Intelligent Systems from the University of Pittsburgh and MS in Computational Linguistics from Carnegie Mellon University. She is a Research Associate at the University of Pittsburgh's

Learning Research and Development Center where she has developed and tested four major dialogue-based learning systems in laboratory and classroom settings. Her current research focus is on adapting content according to user needs with an emphasis on adapting the content presented during tutorial dialogue to match a student's level of mastery.

1 Introduction

One-on-one human tutoring has been shown to be an effective method of instruction (Bloom, 1984). Socio-cognitive theories attempt to explain this success in terms of the interactivity occurring through the dialogue (Chi et al., 2001; Boyer et al., 2010). Although there is abundant empirical evidence that interaction between a student and tutor (or student and peer) supports learning, much less is known about the specific features of effective instructional dialogue. This level of specificity is needed to plan tutorial dialogues in intelligent tutoring systems (ITS).

Researchers in cognitive science and ITS have made significant progress in identifying specific features of human tutorial dialogue that predict learning (Chi et al., 2001, 2008; Forbes-Riley and Litman, 2007; Ward et al., 2009). For example, certain types of discourse relations have been shown to be beneficial for learning (Ward et al., 2009; Katz and Albacete, 2013). Earlier work using lexical cohesive ties was found to be correlated with learning (Ward et al., 2009) and it was found to be possible to identify patterns in a human tutor's use of some ties through the use of machine learning (Katz and Albacete, 2013). Cohesion is considered to be the connectedness of a text (Halliday and Hasan, 1976) and cohesive ties are the various forms of connectedness, such as synonymy and paraphrase. In this paper, we move beyond lexical relations and consider co-constructed discourse relations. Two relations in particular have been found to correlate with learning: tutor generalisation and tutor specification (Ward et al., 2009; Ward and Litman, 2007). Tutor generalisation occurs when the tutor repeats part of a student's utterance, but at a higher level of abstraction. Tutor specification occurs when the tutor repeats part of a student's utterance, but more concretely. Section 2.1 has examples of semantic generalisation and semantic specification.

Inspired by the research above, we aim to develop a dialogue-based tutoring system that will automatically generalise or specify tutor turns, relative to preceding student turns. Our project started with a fully-automatic interactive post-problem reflective dialogue system for physics and modified the reflective dialogue system so that it simulates dialogue decision rules that we predict should improve student learning, relative to a control dialogue system that does not implement these rules. These rules are based on correlational analyses of features of tutorial interaction that predict learning, as measured by pretest to post-test gains. The system has been used to engage in interactive reflective dialogues with high school students after the students have solved introductory physics problems (Katz et al., 2013; Jordan et al., 2013b, 2013a). Previous work has shown that post-problem reflective dialogues are beneficial for student learning (Katz et al., 2003). To achieve the interactivity desired, we must identify *when* a computer tutor should generalise or specify based on when human tutors did so during reflective dialogues. In the current version of our system, this decision is made manually by the authors of the dialogues. The work presented in this paper is the first step in beginning to automate the *when* decision-making.

Other researchers are also developing or evaluating adaptive tutorial dialogue systems. Much of this research focuses on uncovering tutorial dialogue tactics (VanLehn et al., 2003; Pon-Barry et al., 2006). For example, researchers have had success developing tutors that adapt to students' affective states (Forbes-Riley and Litman, 2011, 2012; Aist et al., 2002). Adapting to these affective states involves providing additional feedback from the tutor that addresses the student's affective state (e.g., giving feedback with a positive slant for poorly-performing studious students) (Dennis et al., 2012; Aist et al., 2002; Forbes-Riley and Litman, 2012, 2011). This feedback has led to increased persistence (Aist et al., 2002), learning gains over no feedback on affect (Forbes-Riley and Litman, 2011), and increased speed of learning (Forbes-Riley and Litman, 2009). Detecting these affective states includes such features as body language and facial expressions (D'Mello and Graesser, 2010; Woolf et al., 2009), lexical or dialogue features (D'Mello and Graesser, 2010; Forbes-Riley et al., 2012), response times (Beck, 2004), audio or spoken features (Pon-Barry et al., 2006; Drummond, and Litman, 2010; Forbes-Riley et al., 2012), and student features (Forbes-Riley et al., 2012).

Researchers have also examined adapting pedagogical strategies to individual students. In determining the level of interactivity, it is important to consider the skill level of the student in comparison to the difficulty of the content that is to be learned (VanLehn et al., 2007). For content that is at the current skill level of the student, or easier, high levels of interactivity (e.g., dialogue with tutor) provide no benefit over low or no interactivity (e.g., reading a canned text). For content that is just above the student's skill level (i.e., material the student has not yet mastered, but is ready to learn), then higher levels of interactivity are beneficial (VanLehn et al., 2007). Determining skill level can be gauged with a pretest and monitored through interactions with the tutoring system (Corbett and Anderson, 1994; Pavlik et al., 2009). Interactivity can be at the problem level, where the system will decide which problem to give the student next (Corbett and Anderson, 1994). Finer-grained levels of interactivity decisions (e.g., how intrusive to make hints) can be made using student demographic, temporal, contextual, and performance features (Chi et al., 2010; Chi, 2009; Arroyo et al., 2000).

Students may engage in behaviour that is not conducive to learning, such as 'gaming the system', where students obtain correct answers by taking advantage of the tutoring system's feedback and help (Baker et al., 2004b). Addressing this behaviour has been shown to improve learning (Baker et al., 2006) and has been successfully detected with models using student performance, contextual, and temporal features (Walonoski and Heffernan, 2006; Baker et al., 2004a). However, not all students who take advantage of the tutoring system's help are doing so in a manner that hinders learning (Shih et al., 2008). Distinguishing between helpful and harmful uses of a tutoring system's help can also be done through the use of a model that uses temporal features (Shih et al., 2008).

Finally, prior work on the same data as used in this paper examined lexical changes in abstraction from student turns to tutor turns (Lipschultz et al., 2011). The authors identified important features and feature groups that were helpful in predicting tutor changes in abstraction. For example, they found that information from the dialogues (e.g., which reflection question the student was answering) are most useful for the prediction tasks, and that problem-solving features (e.g., how many mistakes the student made while solving the problem) and student features (e.g., gender) are not as important. This work goes beyond lexical changes in abstraction and looks at semantic changes.

Following from the research above, we aim to develop a tutoring system that automatically adapts tutor turns to generalise or specify relative to preceding student turns. The work presented here takes the first step towards such a system. Others who are also developing interactive tutorial systems have found certain types of features beneficial for identifying when to change the level of interactivity (e.g., adapting to emotional states or eliciting information from a learner versus telling the students that information the information); we used these features as guidance when we selected our features. Research into emotion detection in tutorial dialogue systems found that dialogue context information, such as the number of main questions answered or the number of characters in the student's turn, correlate with various emotional states (D'Mello and Graesser, 2006). Others have found demographic information to be important for determining hint interactivity (Arroyo et al., 2000). Student performance and student dialogue information have been used in research determining when a tutor should elicit information from the student versus telling them (Chi et al., 2011a, 2011b).

We explore how useful each feature is in predicting generalisation and specification by training two decision trees (one for each discourse relation) per feature. In addition to looking at the prediction results, we examine the trees to gain intuitions about why the tutor may have generalised or specified what the student had said. To further investigate feature relationships, we group related features and train decision trees on these feature sets. From these trees, we are able to identify possible rules for why the tutor generalised or specified and the emerging rules suggest plausible explanations for tutor generalisation and specification. Our results suggest that these features may be useful for predicting generalisation and specification and may be helpful for guiding generalisation and specification during natural-language dialogue generation.

2 Corpus

Our corpus is from a previous study (Katz et al., 2003) on the effectiveness of reflection questions after a physics problem solving session with the Andes physics tutoring system (VanLehn et al., 2005). Students taking introductory physics courses at the University of Pittsburgh were recruited. They took a physics pretest, with 9 quantitative and 27 qualitative physics problems. All 36 problems were tagged by physics experts for knowledge components (KCs) that students must have in order to correctly answer the problem. For example, one knowledge component necessary for solving the problem shown in Figure 1 is "Tension in a cord or rope produces a force pulling in toward the center of the cord or rope". Following the pretest, students studied workbook material developed for the experiment and received training on using Andes.

Although there were three conditions in the experiment, we only focus on the Human Feedback condition since we are interested in building more interactive dialogues, which only this condition provides; see Katz et al. (2003) for complete details of all conditions. Students in each condition began by solving a basic mechanics problem in Andes. Each student's tutor observed a student solving the problem from a computer in a remote location, and did not interact with the student. After completing the problem, students in the human feedback condition were presented with several deep-reasoning, 'reflection questions', which they were required to answer before moving on to the next Andes problem. The purpose of the questions was for students to reflect on the problem they had just completed and think about the concepts involved in solving that problem. After typing their answer, students would begin a teletyped dialogue with their human tutor regarding their answer. This dialogue continued until the tutor was satisfied that the student understood the correct answer. Three to eight reflection questions were asked per problem solved in Andes. Each reflection question was tagged by the physics tutors for KCs that the students would need to know to correctly answer the question. There were 12 problems in all. An example problem and the reflection questions associated with it can be found in Figure 1.

Figure 1 Sample problem and the three reflection questions

Andes problem:

A rock climber of mass 55 kg slips while scaling a vertical face. Fortunately, her caribiner holds and she is left hanging at the bottom of her safety line. Find the tension in the safety line.

Reflection questions:

1 What minimum acceleration must the climber have in order for the rope not to break while she is rappelling down the cliff? (You do not have to come up with a numerical answer. Just solve for ‘a’ without any substitution of numbers).

2 Suppose the maximum tension in the rope was 500 N. What would happen to the climber if she hung stationary on the rope?

3 Suppose the climber were rappelling down the rope with a constant velocity equal to or less than the minimum acceleration found in the previous question. Would the rope still break?

After the final problem’s reflection dialogues, students took a post-test that was isomorphic to the pretest and counterbalanced. The study found that students who answered reflection questions learned more than students who did not answer reflection questions. However, there was no significant difference between the human feedback condition and the condition with canned feedback to students’ answers to the reflection questions.

There were 16 students in the human feedback condition (4 male, 12 female). Fifteen students participated in all 60 reflection question dialogues; one only participated in 53, yielding a total of 953 dialogues. There are a total of 2,218 student turns and 2,135 tutor turns in these dialogues. There is an average of 2.32 student turns and 2.24 tutor turns per dialogue. The minimum turns for a dialogue was 1, where the student answered the reflection question correctly and the tutor decided to move on to the next question or problem. Students answered some reflection questions incorrectly, and the tutor then engaged the student in dialogue to correct the student’s answer. The maximum number of dialogue turns was 56.

Each subject was assigned to one of seven tutors. Since there were more students than tutors, some tutors worked with multiple students. No students knew their tutor prior to the study. The tutors had prior experience teaching physics in a classroom or one-on-one tutoring setting; some had done both. Additionally, students who discussed reflection questions with these tutors showed learning gains over the control condition where students did not interact with a tutor and solved more problems instead (Katz et al., 2003). Tutors had opportunities to chat with the subjects before and after tutoring. Thus, the tutors and the subjects had the chance to get to know each other.

2.1 Annotation

This corpus of human feedback condition data has been used in a number of other corpus studies examining the connectedness of text (Ward et al., 2009; Lipschultz et al., 2011; Katz and Albacete, 2013). We use data from the latest study, which examined co-constructed dialogue relations and their correlation with learning (Katz and Albacete, 2013). The reflection dialogue corpus was tagged by human annotators for co-constructed discourse relations. For each turn, the annotators identified segments containing a discourse relation to a segment in the previous speaker's turn, then tagged that segment with the relation identified. The annotators focused on two main discourse relations, generalisation and specification, and various subtypes – for example, the part: whole relation is a type of generalisation.

2.1.1 Generalisation

The generalisation dialogue relation occurs when the second speaker refers to a more general concept, principle, or value than one the first speaker referenced in their preceding turn. For example, in the following exchange, the tutor refers to speed and the student classifies speed as a scalar quantity (italics are added to highlight the relation):

> "**Tutor:** Since the question asked about *SPEED*, suppose we had found v_y to be negative. Should we include the minus sign when giving the speed?
>
> **Student:** I would say no because *speed is scalar* and doesn't include direction."

Generalisation can also occur when the second speaker refers to a physics principle that explains, or is illustrated by, problem-specific content in the first speaker's turn. For example, in the following exchange, the student explains her answer and the tutor offers the general principle about the relationship between acceleration and velocity when an object is slowing down:

> "**Reflection question:** The bullet is traveling to the right. What direction is its acceleration?
>
> **Student:** *to the left because it is making the bullet slow down*
>
> **Tutor:** Good–*when something is slowing down, its acceleration has a component opposite to its velocity*."

2.1.2 Specification

The specification cohesive tie is the opposite of generalisation. It can occur when the second speaker refers to a more specific concept, principle, or value than the one that the first speaker referred to. For example, in the following exchange, the tutor asks for the forces on a climber, and the student names two types of forces:

> "**Tutor:** What are the *forces* on her?
>
> **Student:** her *weight* and the *tension* of the rope."

Specification can also occur when the second speaker instantiates a principle or concept that the first speaker refers to. For example, in the following exchange, the student carries out the tutor's directive to apply Newton's Second Law to the current problem:

> "**Tutor:** Now use *Newton's Second Law* and find [the climber's] acceleration–a number and units; show me the symbols (the algebra).
>
> **Student:** *39/55 = a, a = .71 m/s^2 downwards*."

2.1.3 Tagging

Two annotators were each assigned a subset of the data to reduce duplicate tagging. The subsets were such that, combined, the entire data set was annotated. The annotators tagged segments of students' and tutors' dialogue turns for generalisation and specification. In addition, they tagged for particular types of generalisation and specification based on Rhetorical Structure Theory (Mann and Thompson, 1988), such as part:whole relations or member:set relations. See Katz and Albacete (2013) for more detail. They then checked the other annotator's tags. Any disagreements were discussed and reconciled. Of the 2,135 tutor turns, 141 contained generalisation and 132 contained specification. While it is possible for a tutor to both specialise and generalise in a single turn, our corpus does not contain any instances.

We use this tagged corpus to build models predicting the tutor's next turn based on features described in Section 3. We focus on predicting when the tutor changed level of abstraction because, as we design a computer tutor, we have the ability to modify the computer tutor's level of abstraction. Since students' change in level of abstraction is harder to control, this is left for future work. In Section 4, we discuss how models were trained and evaluated to predict tutor generalisation and tutor specification. Due to the small data size, we consider these models to only offer recommendations of when it might be best for a computer tutor to generalise or specify.

3 Features

From the data in our corpus, we identified features that are either easily extracted or readily available from a fully-automated dialogue-based tutoring system. Similar features have been used in previous work on emotion detection in tutorial dialogue systems (D'Mello and Graesser, 2006), determining hint interactivity (Arroyo et al., 2000), and research on determining when a tutor should elicit information from the student or give them information (Chi, 2009). The features we identified were partitioned into three sets based on the source of the feature: student, problem, and reflection dialogue. This allowed us to explore not only which features are useful, but also which sources are most useful for predicting tutor generalisation or specification.

We performed a median split on most of the numeric features for ease of interpreting the decision trees. Table 1 shows the range and median for all numeric features and the set of values for non-numeric features. Features which have been median-split are indicated by a 'Y' in the binned column.

Table 1 Information on feature values

Set	*Feature*	*Low*	*Median*	*High*	*Binned*	*Non-numeric values*
Student	Gender	-	-	-	N	Female, male
	PreQualScore	0.30	0.70	0.81	Y	
	PreQuantScore	0.00	0.33	0.78	Y	
Problem	NextStepHelp	0	3	52	Y	
	WhatsWrongHelp	0	3	43	Y	
	UnsolicitedHelp	0	4	27	Y	
	NumErr	0	13	72	Y	
	NumCorr	0	19	60	Y	
	NumEntries	0	22	86	Y	
	CorrAns	0	1	3	Y	
	Time2SolveNorm	0.08	0.94	3.06	Y	
Reflection	RQPosition	0.125	0.60	1.0	N	
	PrevRQLength	1	2	33	N	
	Time2AnsNorm	0.00	0.85	136.50	Y	
	TurnPosition	1	11	56	N	
	StuWordCount	0.00	6.00	88.00	Y	
	DomainWord%	0.00	0.08	3.00	Y	
	AvgKCScore	0.00	0.50	1.00	Y	
	CorrectCumulative	0.00	0.74	1.00	Y	
	CorrectPrevRQ	0.00	1.00	1.00	Y	
	CorrectRQCumulative	0.00	0.50	1.00	Y	
	CorrectLast10	0.00	0.75	1.00	Y	

3.1 Student features

The features in this set represent information about the student before tutoring began. Their values remain constant over the course of the entire tutoring session. Below are the three features in this set.

- *Gender:* Female or male.
- *PreQualScore:* Score on the qualitative part of the pretest (median split: high, low).
- *PreQuantScore:* Score on the quantitative part of the pretest (high, low).

3.2 Problem features

The features in this set come from the problem-solving sessions in Andes. The values are specific to each problem and subsequent reflection dialogue; they are reset at the start of the next problem. Since problem solving completes before the reflection discussion begins, the values remain constant for all reflection dialogues for that problem. Tutors observed the students solving the problems in Andes and so would have been aware of the approximate values of these variables. Future ITSs would also have access to this information in real time.

- *NextStepHelp:* How often student requested help from Andes on what step to do next (high, low).
- *WhatsWrongHelp:* How often student asked Andes what was wrong with their work (high, low).
- *UnsolicitedHelp:* How often Andes offered an unsolicited hint (high, low).
- *NumErr:* Number of incorrect student entries during problem-solving process (high, low).
- *NumCorr:* Number of correct student entries during problem-solving process (high, low).
- *NumEntries:* Total number of student entries in the interface (sum of NumErr and NumCorr) (high, low).
- *CorrAns:* Total number of correct answers entered (not intermediate entries) by student (high, low).
- *Time2SolveNorm:* Time (in seconds) student spent solving the problem, divided by the average time spent solving this problem by the students in the other conditions of the study (slow, fast).

3.3 Reflection features

The features in this set come from the dialogues for each of the reflection questions, so the values are specific to the dialogue for each reflection question. Since we will be predicting generalisation and specification during the reflection dialogues, some of these values will change over the course of each reflection dialogue.

- *RQPosition:* Which reflection question the student is currently discussing for the particular problem, normalised by the number of reflection questions for the problem. We normalise because problems have different numbers of reflection questions; normalising better reflects a student's progress through the post-problem dialogue.
- *PrevRQLength*: Number of turns in the previous reflection question's dialogue.
- *Time2AnsNorm*: How long (in seconds) it took for the student to respond to the tutor's previous message, normalised by the number of characters in the student's response (slow, fast).
- *TurnPosition*: Position in the reflection question dialogue.
- *StuWordCount*: Count of words in the student's preceding turn (high, low).
- *DomainWord%*: Of all words in student's preceding turn, the percentage that are physics domain words (from: http://scienceworld.wolfram.com/physics/letters/) (high, low).
- *AvgKCScore*: For the KCs required to correctly answer the reflection question, the student's average score on the pretest problems also requiring those KCs (high, low).

- *CorrectCumulative*: For all preceding turns, percent of correct student responses divided by sum of correct and incorrect student responses (high, low).
- *CorrectPrevRQ*: From only the immediately preceding reflection dialogue, percent of correct student responses divided by sum of correct and incorrect student responses (high, low).
- *CorrectRQCumulative*: From all preceding turns in the current reflection dialogue, percent of correct student responses divided by sum of correct and incorrect student responses (high, low).
- *CorrectLast10*: From the last ten turns, percent of correct student responses divided by sum of correct and incorrect student responses (high, low).

4 Machine learning

As mentioned above, we are interested in using this corpus to predict when a computer-based tutor might generalise in a post-problem reflective dialogue and when such a tutor might specify. Thus, we will be building from this corpus two models, one to predict when the human tutors generalised and the other to predict when the human tutors specified. The tags from Katz and Albacete (2013) were done on segments of a turn. In this work, we predict at the turn level, so if any segment of a turn was labelled as generalisation or specification, then the turn was labelled as generalisation or specification. Since predictions are at the turn level, the segment-level tags were propagated to the turn level. Both of these prediction tasks are binary classifications, with yes meaning that the tutor provided a generalisation or specification from the student's turn preceding the turn we are attempting to predict and *no* meaning that the tutor did not.

Since the original data has a large bias towards not generalising and towards not specifying, we use WEKA's CostSensitiveClassifier (Hall et al., 2009) to increase the cost of misclassifying *yes* as *no*. We choose this method over others for handling bias, such as downsampling, because this method allows us to use the full data set for machine learning. The machine learning algorithm used to perform the classification is J48 Decision Trees, WEKA's implementation of the C4.5 decision tree algorithm. We chose this algorithm because it allows us to easily see relationships between features used in the trees. The default settings for the decision tree were modified to have a confidence factor of 0.125 and minimum number of instances in the leaf nodes to 60. Both settings were modified to encourage shorter trees, to ease interpretation of relationships. There was no significant difference between these trees and the trees using the default settings.

We compare the performance of our models to a majority class baseline. The majority class baseline for both tasks predicts *no*. For each model, we performed leave-one-student-out cross-validation. As stated earlier, since there were not many tutors, the models learned only indicate when a computer tutor may want to change the level of abstraction. We leave as future work studying a larger corpus.

4.1 Single-feature trees

We begin by examining how useful each feature is in isolation. For each classification task, we trained one decision tree for each of the features listed in Section 3. As we are not specifically interested in optimising for precision or recall, we rank performance by F1. In this context, ‘precision’ for generalisation refers to how many true tutor generalisations there were out of all the tutor turns the model predicted were generalisations. ‘Recall’ for generalisation is how many generalisations were correctly identified by the model, out of all generalisation tutor turns in the corpus. Similar definitions are used for specification.

Since the data is heavily skewed and we are interested in the minority class, we report the precision, recall, and F1 for the class of interest (*yes*). Table 2 shows the baseline and each of the models.

Others have suggested looking at the unweighted average precision, recall, and F1 when evaluating the performance of models in situations with a large skew in the classes (Schuller et al., 2009). These metrics consider both classes in the results, rather than just the class of interest. Thus, we also present the unweighted average for the baseline and each of the models in Table 3. Although we will use Table 2 for selecting the best models, we obtained similar results from the unweighted average evaluation method.

In this section, we focus solely on the single feature models.

4.1.1 Generalisation

Two of the models using only Student features – Gender and PreQualScore – are significantly better than baseline for F1. Of the two, PreQualScore performs significantly better. Since this feature represents how well the student performed on the qualitative, or conceptual, questions on the pretest, it is a good indication of how well the student understood the concept. The decision tree indicates that when the PreQualScore is low, predict generalisation, and when it is high, predict no generalisation. This suggests that tutors tend to generalise when students have poor conceptual understanding. Perhaps tutors are showing how instantiation of one or more concepts in the current problem connects to a more general physics concept. Since this feature is constant throughout an entire tutoring session, this model will make the same prediction for all tutor turns. That is, it adapts to the student’s incoming qualitative knowledge, but to nothing during tutoring, despite making predictions at the turn level. Therefore, it will make the same prediction for each tutor turn in the dialogue.

Of the eight models using only one Problem feature each, five performed significantly better than baseline for F1. The best feature in this group appears to be UnsolicitedHelp, which is a median split of the count of the number of times Andes provided unsolicited help; however, UnsolicitedHelp is not significantly better than Time2SolveNorm. The decision tree indicates that when UnsolicitedHelp is high, human tutors tended to generalise, and when it is low, human tutors tended to not generalise. Receiving unsolicited help on the steps needed to solve a physics problem indicates that the student does not know how to solve it. Therefore, the human tutor may have been trying to explain general concepts to the student so that the knowledge can transfer to another problem that uses the same concepts. Since this feature is constant throughout the reflection dialogues for a given problem, this model will make the same prediction for all tutor turns within those reflection dialogues. Therefore, it adapts to students’ performance during problem solving, but to nothing during the discussions, despite making predictions at the turn level.

Table 2 Comparing feature sets across the *yes* metrics for both generalisation and specification classification tasks

		Generalisation			*Specification*		
		Precision	*Recall*	*F1*	*Precision*	*Recall*	*F1*
Baseline		0.065	0.461	0.113	0.061	0.493	0.108
Single features	*Student*						
	Gender	0.068	0.747	*0.125*	*0.071*	*0.863*	*0.13*
	PreQualScore	*0.079*	*0.68*	*0.142*	*0.067*	*0.583*	*0.12*
	PreQuantScore	0.064	0.496	0.113	0.058	0.478	0.104
	Problem						
	NextStepHelp	*0.071*	*0.614*	*0.128*	0.058	0.538	0.105
	WhatsWrongHelp	0.062	0.5	0.11	0.058	0.529	0.104
	UnsolicitedHelp	*0.076*	*0.631*	*0.135*	*0.068*	*0.614*	*0.123*
	NumErr	0.059	0.47	0.105	*0.072*	*0.629*	*0.129*
	NumCorr	0.071	*0.546*	*0.125*	0.06	0.487	0.106
	NumEntries	0.068	*0.535*	0.121	*0.074*	*0.629*	*0.132*
	CorrAns	*0.071*	*0.672*	*0.129*	*0.068*	*0.069*	*0.124*
	Time2SolveNorm	*0.074*	*0.582*	*0.131*	0.056	0.463	0.1
	Reflection						
	RQPosition	*0.079*	*0.535*	*0.138*	0.055	0.487	0.099
	PrevRQLength	0.062	0.382	0.107	0.061	*0.672*	*0.111*
	Time2AnsNorm	*0.1*	*0.745*	*0.176*	*0.086*	*0.689*	*0.153*
	TurnPosition	0.061	0.517	0.11	*0.067*	*0.518*	*0.119*
	StuWordCount	0.06	0.452	0.105	0.057	0.436	0.101
	DomainWord%	0.064	0.495	0.113	*0.068*	*0.587*	*0.121*
	AvgKCScore	*0.069*	*0.709*	*0.126*	0.057	0.335	0.097
	CorrectCumulative	0.061	0.457	0.108	*0.066*	0.51	*0.118*
	CorrectPrevRQ	*0.075*	*0.505*	*0.131*	0.056	0.468	0.101
	CorrectRQCumulative	0.063	0.427	0.11	0.053	0.411	0.093
	CorrectLast10	*0.071*	*0.535*	*0.126*	0.056	0.44	0.099
Feature sets							
	Student	*0.077*	*0.586*	*0.136*	*0.082*	*0.725*	*0.147*
	Problem	0.068	0.427	0.117	*0.073*	*0.641*	*0.131*
	Reflection	0.108	*0.597*	*0.182*	*0.091*	*0.518*	*0.156*
Aggregated feature sets							
	StudentProblem	*0.076*	0.466	*0.13*	*0.09*	*0.641*	0.158
	StudentReflection	0.108	*0.609*	0.184	*0.092*	*0.555*	*0.158*
	ProblemReflection	*0.1*	*0.587*	*0.171*	0.094	0.491	*0.158*
	Overall	*0.103*	*0.615*	*0.176*	*0.088*	0.495	*0.149*

Notes: Italicised values indicate results significantly better than baseline (p < 0:05). All other values are not significantly different from the baseline. The underlined values are the greatest in that column.

Table 3 Comparing feature sets across the unweighted average metrics for both generalisation and specification classification tasks

		Generalisation			*Specification*		
		Precision	*Recall*	*F1*	*Precision*	*Recall*	*F1*
Baseline		0.497	0.491	0.494	0.497	0.496	0.497
Single features	*Student*						
	Gender	*0.504*	*0.511*	*0.508*	*0.517*	*0.551*	*0.534*
	PreQualScore	*0.514*	*0.555*	*0.534*	*0.505*	*0.521*	*0.513*
	PreQuantScore	0.497	0.488	0.492	0.495	0.481	0.488
	Problem						
	NextStepHelp	*0.505*	*0.519*	*0.512*	0.495	0.480	0.487
	WhatsWrongHelp	0.495	0.48	0.487	0.495	0.482	0.489
	UnsolicitedHelp	*0.510*	*0.538*	*0.524*	*0.507*	*0.528*	*0.517*
	NumErr	0.492	0.471	0.482	*0.510*	*0.542*	*0.526*
	NumCorr	*0.503*	*0.512*	*0.507*	0.498	0.493	0.496
	NumEntries	*0.501*	*0.504*	*0.503*	*0.512*	*0.550*	*0.530*
	CorrAns	*0.506*	*0.524*	*0.515*	*0.507*	*0.527*	*0.517*
	Time2SolveNorm	*0.507*	*0.529*	*0.518*	*0.493*	0.472	0.482
	Reflection						
	RQPosition	*0.513*	*0.550*	*0.531*	0.491	0.470	0.480
	Time2AnsNorm	0.533	0.631	0.578	*0.524*	*0.601*	*0.560*
	TurnPosition	0.496	0.486	0.491	*0.503*	*0.514*	*0.509*
	StuWordCount	0.492	0.474	0.483	0.495	0.481	0.488
	DomainWord%	0.495	0.481	0.488	*0.506*	*0.525*	*0.516*
	AvgKCScore	*0.505*	*0.516*	*0.511*	0.494	0.481	0.487
	CorrectCumulative	0.493	0.474	0.483	*0.504*	*0.517*	*0.510*
	CorrectPrevRQ	*0.507*	*0.528*	*0.517*	0.492	0.471	0.481
	CorrectRQCumulative	0.497	0.487	0.492	0.491	0.462	0.476
	CorrectLast10	*0.504*	*0.516*	*0.510*	0.492	0.469	0.480
	Feature sets						
	Student	*0.510*	*0.523*	*0.524*	*0.522*	*0.592*	*0.555*
	Problem	*0.502*	*0.509*	*0.506*	*0.514*	*0.557*	*0.534*
	Reflection	*0.526*	*0.598*	*0.560*	*0.513*	*0.553*	*0.532*
	Aggregated feature sets						
	StudentProblem	*0.506*	*0.523*	*0.514*	0.525	0.606	0.563
	StudentReflection	*0.528*	*0.606*	*0.564*	*0.519*	*0.578*	*0.547*
	ProblemReflection	*0.521*	*0.579*	*0.549*	*0.515*	*0.560*	*0.537*
	Overall	*0.521*	*0.579*	*0.548*	*0.516*	*0.565*	*0.540*

Notes: Italicised values indicate results significantly better than baseline (p < 0:05). All other values are not significantly different from the baseline. The underlined values are the greatest in that column.

Of the 11 models using only one Reflection feature each, five performed significantly better than baseline. Time2AnsNorm performed significantly better than the other four and the decision tree indicates that when the student was fast to respond, the tutor would generalise. As we will see below, the tutor also tends to specify when the student responds quickly, so interpretation of this decision tree is unclear.

Overall, it appears that tutors tend to generalise when students display poor understanding of a concept, in the case at hand.

4.1.2 Specification

Two of the three models using only Student features – Gender and PreQualScore – are significantly better than baseline. Of the two, Gender performs better, although not statistically significantly, and its tree indicates that the tutors tend to specify when the student is female. It is unclear how gender is involved in the tutor's decision to specify. The tutors were aware of the gender of the student because they chatted with the students before and after tutoring. It is possible that Gender represents an attribute or set of attributes about students that could influence a tutor's decision to specify, such as incoming physics knowledge. We examined whether there was a significant difference between females and males on pretest score, quantitative-only pretest scores, and qualitative-only pretest scores, and found that there was no significant difference on any of these scores ($p \geq 0.92$). In future work, we plan on examining other potential gender differences, such as problem-solving behaviour or dialogue differences (e.g., word choice). It may also be the case that the chatting before and after tutoring sessions led to tutor-student rapport. The tutors may have been more likely to go into more detail with students they had better rapport with.

Four of the eight models using only one Problem feature performed significantly better than baseline. Of these five, NumEntries performed best, although not significantly better than NumErr. NumEntries' decision tree indicates that when the student had many entries in Andes, the tutor was more likely to specify. More entries in Andes – vectors, scalars, and equations – tended to indicate students were trying whatever they could to solve the problem, usually on their own. So, perhaps the tutor specified to give the student very specific explanations and instructions to help the student apply the abstract physics concepts to the specific situation presented in the problem at hand. The NumErr tree supports this interpretation. For students with an above-median number of errors during problem-solving, the tutors may specify more.

Of the 11 models using only one reflection feature each, five performed significantly better than baseline. Of these five, Time2AnsNorm performed best; its decision tree indicates that when the student was fast to respond, the tutor would specify. Since the tutor also tends to generalise in these cases, it is not clear how a computer tutor can use this feature to decide whether to generalise or specify. Although it is possible for the human tutor to have generalised and specified in the same turn, our corpus shows no instances of the tutor doing this.

Overall, interpretation of the one-feature specification trees is generally not clear. From the Problem feature set, it appears that the tutor may specify to help the student apply the physics concepts to the specific situation presented in the problem.

4.2 Best-performing trees

Having examined how important individual features are, we now examine the best-performing trees for each classification task, where "best" is determined by how well the models performed on the *yes* class. While the best trees perform significantly better than baseline, their performance is still low. Recall that there is a large bias towards not generalising and towards not specifying. For the work presented here, we are interested in testing the feasibility of the classification tasks and in identifying patterns that predict tutor generalisation and specification. In future work, we propose improving classification performance. However, when looking at Table 3, which considers performance on both classes, we see that the numbers do improve.

4.2.1 Generalisation

For predicting when the tutors tended to generalise, the best model is the StudentReflection model, which had the highest F1, although not significantly different than Time2AnsNorm, Reflection, ProblemReflection, or Overall. The StudentReflection model can be seen in Figure 2.

If the student is slow to respond and is working on an early reflection dialogue (RQPosition <= 0.143), then the tutor will tend to generalise. Being slow to respond could indicate that the student does not have all the necessary knowledge to answer the question or is learning to link the knowledge required to answer the question. Hence, the answer provided by the student may be incomplete, not quite coherent, or too specific to the problem at hand. Therefore the tutor may try to complete the given answer or explain it at a slightly higher level.

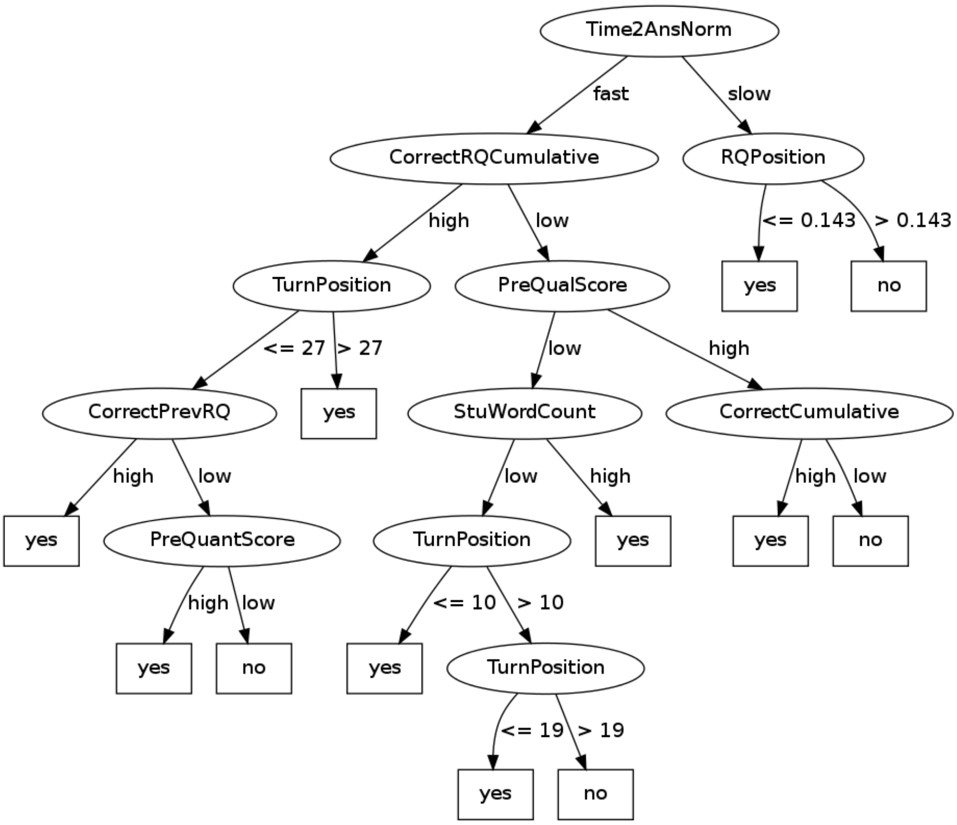
If the student responds quickly, then the tutors tend to generalise for a variety of reasons.

First, there are instances of when the tutor may generalise when the student appears to be showing evidence of doing well. We see that when the student responds fast, CorrectRQCumulative is high, and either CorrectPrevRQ is high or PreQuantScore is high.

Second, if the student has been doing well overall (PreQualScore = high and CorrectCumulative = high), but is struggling with the current dialogue (CorrectRQCumulative = low), then the tutors tend to generalise. The tutor might do this to connect the details of this dialogue to the concepts in previous dialogues or problems.

Finally, the tutor may also generalise if the student appears to be struggling. When the student is getting a lot wrong in the current dialogue (CorrectRQCumulative = low), and performed poorly on the pretest's qualitative questions (PreQualScore = low), then the tutors tend to generalise. Here, the tutor may be trying to explain concepts before going on to the specifics of the reflection question. Another possibility is that the tutor is explaining the general line of reasoning before explaining each specific step in it.

Figure 2 Decision tree to predict generalisation using the combined feature sets of student and reflection



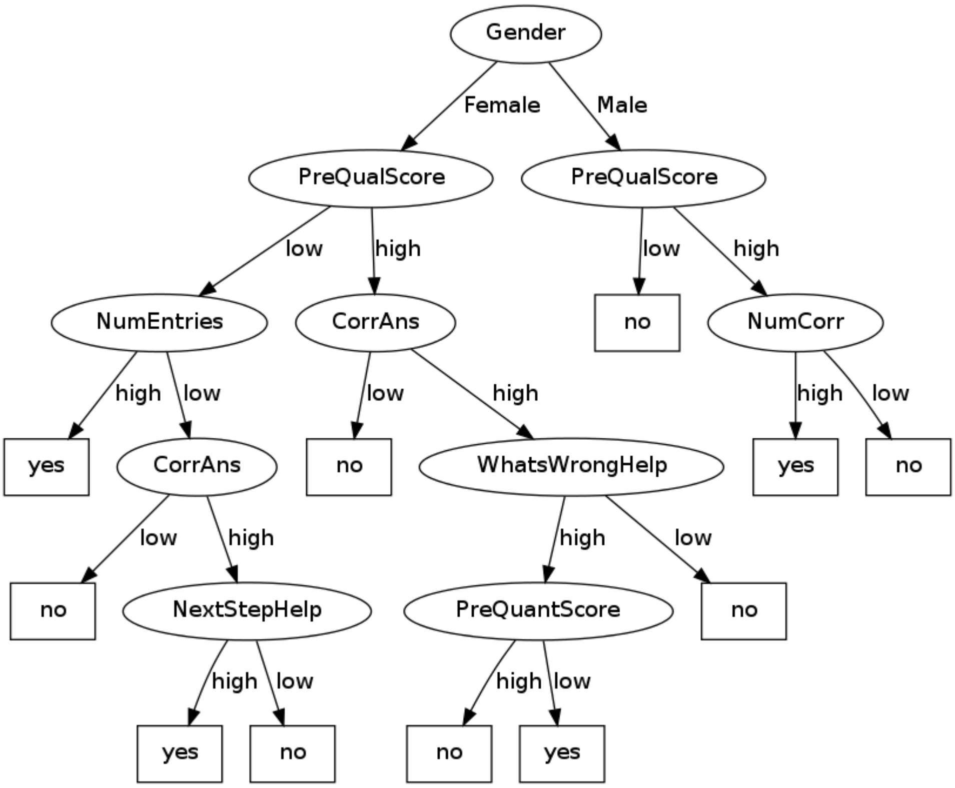
4.2.2 Specification

For predicting when the tutor may specify, the best model is the one that uses the Student and Problem feature sets, although it is not significantly different than Time2AnsNorm, Student, Reflection, StudentReflection, ProblemReflection, or Overall. The decision tree that is learned using these features can be seen in Figure 3.

We see that when the student shows evidence of understanding, the tutor may specify more often. This occurs for males when their qualitative pretest scores are high and have many correct entries in Andes. This occurs for females when their qualitative pretest scores are high, have many correct answers in Andes, and did not have many What's wrong? help requests in Andes. This may be happening when students are doing well and show understanding of the concepts; then the tutor may start focusing on details so that students' answers are complete and precise.

When students show evidence of struggling with problem solving, the tutors tend to specify. When females asked many "what's wrong" questions during problem solving (WhatsWrongHelp = high) and had a low score on the quantitative portion of the pretest, then the tutor specified. Or, the tutor may have been trying to explain concepts by using specific examples from the problem. It could also be the case that because the student did not show much mathematical ability (low score on the quantitative portion of the pretest) that the tutor may be asking the student to instantiate all equations that are relevant to the problem at hand.

Figure 3 Decision tree to predict specification using the combined feature sets of student and problem



5 Discussion

This work builds upon previous work which showed that particular types of generalisation and specification discourse relations predict learning (Katz and Albacete, 2013). The goal of the current analysis was to determine whether features and feature groups that can be automatically extracted from Andes tutoring logs and reflection dialogue logs may be useful for predicting whether a tutor may generalise or specify. We examined a corpus of post-problem human-human tutorial dialogues of conceptual physics to find instances of when the tutor would generalise or specify over the student's preceding turn. We then used decision trees both to identify useful features and feature sets, and to identify patterns between features and tutor generalisation and specification. Finally, we offered interpretations on why the tutor may have followed these patterns.

The most important feature for both the generalisation task and the specification task was how long it took the student to respond to the tutor's question during reflection. In both cases, the longer the student took, the more likely the tutor was to generalise and the more likely they were to specify. Since the models predict that a computer tutor should specify and generalise on the same turn, but we do not find any instances of these in the data, this feature alone is not enough to model when the tutor should specify and generalise. So, we explored more complex models that used multiple features.

For the generalisation task, we find that the model with the highest F1 uses a combination of the student and reflection features. The tutor generalised for students struggling as well as students not struggling. The tutor may wish to generalise for struggling students to introduce general concepts when the student is trying to discuss specifics of a problem. For students not struggling, the tutor may want the student to talk about general concepts because thinking in terms of general concepts may help the knowledge transfer to future problems or reflection questions.

For the specification task, we find that the best model uses both the Student and Problem feature sets. As with generalisation, we found that tutors specify for struggling students and for students not struggling. When a student is struggling, the tutor may speak concretely to ground the discussion in a specific situation to help the student understand the concepts. When students show evidence of understanding, the tutor appears to test their knowledge or try to prepare them for upcoming problems by asking specification questions. It is also possible that the tutor specifies because the student understands the basic concepts and is ready to discuss more details about those concepts.

We also found that gender is an important feature, particularly for the specification task, but it is unclear why. The tutor was aware of the student's gender, so the tutor may have unknowingly varied behaviour based on the student's gender. Another possibility is that males and females think about or speak about physics differently. For example, males may speak in more specific terms, causing the tutor to more often generalise. To investigate the latter possibility on the current corpus, correlational analysis should be performed between gender and the specificity of student turns.

6 Future work

Augmenting a computer tutor to automatically generalise or specify over the student's preceding turn presents many interesting research challenges. While many models in this paper performed significantly better than baseline, there is still much room for improvement in learning models. Therefore, one challenge is improving classification. In this paper, we chose decision trees because they allow for easy interpretation of feature relationships, but other classification algorithms may provide significantly better classification results. Additionally, other features, perhaps those that are not easily obtainable from an automatic computer tutor, such as other dialogue relations, may improve classification performance. Furthermore, different types of generalisation and specification may have different learning models. For example, predicting part:whole relations might be different from predicting instance:abstract relations. Instead of learning one model for generalisation and one for specification, better performance might be achieved by learning one model for each type. These are each avenues for improving classification performance.

Another challenge is to identify what student spans within a dialogue turn to generalise or specify over. The analysis in this paper focused on identifying whether or not the tutor should generalise or specify. But, to augment a computer tutor to automatically generalise or specify, it must be able to identify the correct span of the student's turn. Therefore, an additional classifier must be developed to identify those spans.

Once the span of the student's turn is identified, the tutor must then generalise or specify over that span. While lexical databases, such as WordNet (Fellbaum, 2010), provide hypernym relations, they contain scientific inaccuracies (Lipschultz and Litman, 2010). Additionally, the lexical database would provide lexical changes in abstraction, but creating semantic changes in abstraction will require additional work. Finally, it is not yet known how many levels of abstraction the tutor changed, nor why the tutor changed that many levels. Therefore, a classifier must be developed to determine how many levels of abstraction to change. Additionally, a semantic hierarchy should be developed.

As noted above, the tutor specified based on gender, but it is unclear why gender was influential in the tutor's decision. We suspect that the tutor did not rely on gender, but rather other variables which correlate with gender that we did not consider. In this work, we tested whether gender correlated with pretest scores and found that they did not. In future work, we propose testing other relationships, such as problem-solving behaviour, dialogue acts, or other measures of incoming knowledge.

This paper explored tutor changes in abstraction because it is easier to control in a computer tutoring system than student shifts in level of abstraction. However, performing similar analysis on student changes can offer insights into when a student might generalise or specify. Findings from that analysis can then be used in a computer tutoring system to encourage future students to generalise or specify. Therefore, we propose as future work developing models of student generalisation and specification.

Finally, the research presented in this paper used a small group of tutors. Conclusions drawn from this sample only indicate when a computer-based tutor may want to generalise. Studying a larger group of skilled tutors' behaviour to determine when they tend to generalise and specialise can provide a better idea of when it is best for a computer-based tutor to generalise or specialise.

Acknowledgements

The authors would like to thank the other members of the Rimac project team for their contributions: Michael Ford, Scott Silliman, Christine Wilson, Stefani Allegretti, and Kevin Krost. This research was supported by the Institute of Education Sciences, US Department of Education, through Grant R305A10063 to the University of Pittsburgh. The opinions expressed are those of the authors and do not represent the views of the Institute or the US Department of Education.

References

Aist, G., Kort, B., Reilly, R., Mostow, J. and Picard, R. (2002) 'Experimentally augmenting an intelligent tutoring system with human-supplied capabilities: adding human-provided emotional scaffolding to an automated reading tutor that listens', *Proceedings of the Intelligent Tutoring Systems Conference 2002 Workshop on Empirical Methods for Tutorial Dialogue Systems*, pp.483–490.

Arroyo, I., Beck, J., Woolf, B., Beal, C. and Schultz, K. (2000) Macroadapting animalwatch to gender and cognitive differences with respect to hint interactivity and symbolism', in

Gauthier, G., Frasson, C. and Van Lehn, K. (Eds.): *Intelligent Tutoring Systems*, *Volume of LNCS*, pp.574–583, Springer, Berlin/Heidelberg.

Baker, R., Corbett, A., Koedinger, K., Evenson, S., Roll, I., Wagner, A., Naim, M., Raspat, J., Baker, D. and Beck, J. (2006) 'Adapting to when students game an intelligent tutoring system', *Proceedings of the Intelligent Tutoring Systems Conference*, Springer, pp.392–401.

Baker, R.S., Corbett, A.T. and Koedinger, K.R. (2004a) 'Detecting student misuse of intelligent tutoring systems', *Proceedings of the Intelligent Tutoring Systems Conference*, Springer, pp.54–76.

Baker, R.S., Corbett, A.T., Koedinger, K.R. and Wagner, A.Z. (2004b) 'Off-task behavior in the cognitive tutor classroom: when students game the system', *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, pp.383–390.

Beck, J.E. (2004) 'Using response times to model student disengagement', *Proceedings of the ITS2004 Workshop on Social and Emotional Intelligence in Learning Environments*, pp.13–20.

Bloom, B.S. (1984) 'The 2 Sigma problem: the search for methods of group instruction as effective as one-to-one tutoring', *Educational Researcher*, Vol. 13, No. 6, pp.4–16.

Boyer, K., Phillips, R., Ingram, A., Ha, E., Wallis, M., Vouk, M. and Lester, J. (2010) 'Characterizing the effectiveness of tutorial dialogue with hidden Markov models', *Proceedings of the Intelligent Tutoring Systems Conference*, pp.55–64.

Chi, M. (2009) *Do Micro-Level Tutorial Decisions Matter: Applying Reinforcement Learning to Induce Pedagogical Tutorial Tactics*, PhD dissertation, University of Pittsburgh, Intelligent Systems Program, November.

Chi, M., VanLehn, K. and Litman, D. (2010) 'Do micro-level tutorial decisions matter: applying reinforcement learning to induce pedagogical tutorial tactics', *Proceedings of the Intelligent Tutoring Systems Conference*, Springer, pp.224–234.

Chi, M., VanLehn, K., Litman, D. and Jordan, P. (2011a) 'An evaluation of pedagogical tutorial tactics for a natural language tutoring system: a reinforcement learning approach', *International Journal of Artificial Intelligence in Education*, Vol. 21, No. 2, pp.83–113.

Chi, M., VanLehn, K., Litman, D. and Jordan, P. (2011b) 'Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies', *Proceedings of the User Modelling and User-Adapted Interaction Conference*, Vol. 21, Nos. 1–2, pp.137–180.

Chi, M.T.H., Roy, M. and Hausmann, R.G.M. (2008) 'Observing tutorial dialogues collaboratively: insights about human tutoring effectiveness from vicarious learning', *Cognitive Science*, Vol. 32, No. 2, pp.301–341.

Chi, M.T.H., Siler, S.A., Jeong, H., Yamauchi, T. and Hausmann, R.G. (2001) 'Learning from human tutoring', *Cognitive Science*, Vol. 25, No. 4, pp.471–533.

Corbett, A.T. and Anderson, J.R. (1994) 'Knowledge tracing: modeling the acquisition of procedural knowledge', *Proceedings of the User Modelling and User-Adapted Interaction Conference*, Vol. 4, No. 4, pp.253–278.

D'Mello, S. and Graesser, A. (2006) 'Affect detection from human-computer dialogue with an intelligent tutoring system', *Intelligent Virtual Agents*, pp.54–67, Springer.

D'Mello, S.K. and Graesser, A. (2010) 'Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features', *User Modeling and User-Adapted Interaction*, Vol. 20, No. 2, pp.147–187.

Dennis, M., Masthoff, J. and Mellish, C. (2012) 'Adapting performance feedback to a learner's conscientiousness', *Proceedings of the User Modelling and User-Adapted Interaction Conference*, pp.297–302.

Drummond, J. and Litman, D. (2010) 'In the zone: towards detecting student zoning out using supervised machine learning', *Intelligent Tutoring Systems*, pp.306–308, Springer.

Fellbaum, C. (2010) 'Wordnet', *Theory and Applications of Ontology: Computer Applications*, Vol. 38, No. 11, pp.231–243.

Forbes-Riley, K. and Litman, D. (2007) 'Investigating human tutor responses to student uncertainty for adaptive system development', *Proceedings of the 2nd International Conference on Affective Computing and Intelligent Interaction*, pp.678–689.

Forbes-Riley, K. and Litman, D. (2009) 'Adapting to student uncertainty improves tutoring dialogues', *Proceeding of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*, pp.33–40.

Forbes-Riley, K. and Litman, D. (2011) 'Designing and evaluating a wizarded uncertainty-adaptive spoken dialogue tutoring system', *Computer Speech & Language*, Vol. 25, No. 1, pp.105–126.

Forbes-Riley, K. and Litman, D. (2012) 'Adapting to multiple affective states in spoken dialogue', *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pp.217–226.

Forbes-Riley, K., Litman, D., Friedberg, H. and Drummond, J. (2012) 'Intrinsic and extrinsic evaluation of an automatic user disengagement detector for an uncertainty-adaptive spoken dialogue system', *Proc. NAACL-HLT*.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I.H. (2009) 'The WEKA data mining software: an update', *ACM SIGKDD Explorations Newsletter*, Vol. 11, No. 1, pp.10–18.

Halliday, M.A.K. and Hasan, R. (1976) *Cohesion in English*, Longman, London.

Jordan, P., Albacete, P., Ford, M.J., Katz, S. and Lipschultz, M. (2013a) 'Eliciting student explanations during tutorial dialogue for the purpose of providing formative feedback', *Artificial Intelligence in Education Workshop on Formative Feedback in Interactive Learning Environments*.

Jordan, P., Albacete, P., Ford, M.J., Katz, S., Lipschultz, M., Litman, D., Silliman, S. and Wilson, C. (2013b) 'The Rimac tutor – a simulation of the highly interactive nature of human tutorial dialogue: an interactive event', *Artificial Intelligence in Education Conference (AIED)*.

Katz, S. and Albacete, P. (2013) 'A tutoring system that simulates the highly interactive nature of human tutoring', *Journal of Educational Psychology*, Vol. 105, No. 4, Special issue on Advanced Learning Technologies, pp.1126–1141.

Katz, S., Albacete, P., Ford, M.J., Jordan, P., Lipschultz, M., Litman, D., Silliman, S. and Wilson, C. (2013) 'Pilot test of a natural-language tutoring system for physics that simulates the highly interactive nature of human tutoring', *Artificial Intelligence in Education Conference (AIED)*.

Katz, S., Allbritton, D. and Connelly, J. (2003) 'Going beyond the problem given: How human tutors use post-solution discussions to support transfer', *International Journal of Artificial Intelligence in Education*, Vol. 13, No. 1, pp.79–116.

Lipschultz, M. and Litman, D. (2010) 'Correcting scientific knowledge in a general-purpose ontology', *10th International Conference on Intelligent Tutoring Systems (ITS)*.

Lipschultz, M., Litman, D., Jordan, P. and Katz, S. (2011) 'Predicting changes in level of abstraction in tutor responses to students', *Proceedings 24th International FLAIRS (Florida Artificial Intelligence Research Society) Conference*.

Mann, W.C. and Thompson, S. (1988) 'Rhetorical structure theory: toward a functional theory of text organization', *Text*, Vol. 8, No. 3, pp.243–281.

Pavlik, P.I., Cen, H. and Koedinger, K.R. (2009) 'Performance factors analysis–a new alternative to knowledge tracing', *Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*, pp.531–538, IOS Press.

Pon-Barry, H., Schultz, K., Bratt, E.O., Clark, B. and Peters, S. (2006) 'Responding to student uncertainty in spoken tutorial dialogue systems', *International Journal of Artificial Intelligence in Education*, Vol. 16, No. 2, pp.171–194.

Schuller, B., Steidl, S. and Batliner, A. (2009) 'The interspeech 2009 emotion challenge', *Tenth Annual Conference of the International Speech Communication Association*.

Shih, B., Koedinger, K.R. and Scheines, R. (2008) 'A response time model for bottom-out hints as worked examples', *Proceedings of the Educational Data Mining Conference*, p.117.

VanLehn, K., Graesser, A.C., Jackson, G.T., Jordan, P., Olney, A. and Rosé, C.P. (2007) 'When are tutorial dialogues more effective than reading?', *Cognitive Science*, Vol. 31, No. 1, pp.3–62.

VanLehn, K., Lynch, C., Schulze, K., Shapiro, J.A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A. and Wintersgill, M. (2005) 'The Andes physics tutoring system: lessons learned', *International Journal of Artificial Intelligence in Education*, Vol. 15, No. 3, pp.147–204.

VanLehn, K., Siler, S., Murray, C., Yamauchi, T. and Baggett, W.B. (2003) 'Why do only some events cause learning during human tutoring?', *Cognition and Instruction*, Vol. 21, No. 3, pp.209–249.

Walonoski, J. and Heffernan, N. (2006) 'Detection and analysis of off-task gaming behavior in intelligent tutoring systems', *Proceedings of the Intelligent Tutoring Systems Conference*, Springer, pp.382–391.

Ward, A. and Litman, D. (2007) 'Automatically measuring lexical and acoustic/prosodic convergence in tutorial dialog corpora', *Proceedings of the SLaTE Workshop on Speech and Language Technology in Education*.

Ward, A., Connelly, J., Katz, S., Litman, D. and Wilson, C. (2009) 'Cohesion, semantics and learning in reflective dialog', *Proc. AIED Workshop*.

Woolf, B., Dragon, T., Arroyo, I., Cooper, D., Burleson, W. and Muldner, K. (2009) 'Recognizing and responding to student affect', in Jacko, J.A. (Ed.): *Human-Computer Interaction. Ambient, Ubiquitous and Intelligent Interaction, Volume 5612 of Lecture Notes in Computer Science*, pp.713–722, Springer, Berlin Heidelberg.