GRANTEE SUBMISSION REQUIRED FIELDS

Title of article, paper, or other content

Automated Summary Scoring with ReaderBench

All author name(s) and affiliations on PDF. If more than 6 names, ERIC will complete the list from the submitted PDF.

Last Name, First Name	Academic/Organizational Affiliation	ORCID ID
Botarleanu, Robert-Mihai	University Politehnica of Bucharest, Romania	
Dascalu, Mihai	University Politehnica of Bucharest, Romania;	
Allen, Laura K.	University of New Hampshire, Durham, NH, U	
Crossley, Scott Andrew	Department of Applied Linguistics/ESL, Georg	
McNamara, Danielle S.	Department of Psychology, Arizona State Univ	

Publication/Completion Date—(if *In Press,* enter year accepted or completed) 09 July 2021

Check type of content being submitted and complete one of the following in the box below:

- □ If article: Name of journal, volume, and issue number if available
- □ If paper: Name of conference, date of conference, and place of conference
- ☑ If book chapter: Title of book, page range, publisher name and location
- □ If book: Publisher name and location
- □ If dissertation: Name of institution, type of degree, and department granting degree

ITS 2021: Intelligent Tutoring Systems, pp. 321-332, Springer, Cham Switzerland

DOI or URL to published work (if available) https://doi.org/10.1007/978-3-030-80421-3_35

Acknowledgement of Funding— Grantees should check with their grant officer for the preferred wording to acknowledge funding. If the grant officer does not have a preference, grantees can use this suggested wording (adjust wording if multiple grants are to be acknowledged). Fill in Department of Education funding office, grant number, and name of grant recipient institution or organization.

"This work was supported by U.S. Department of Education **[Office name]** National Center for Educat through **[Grant number]** R305A1900 to **Institution]** Arizona State U. The opinions expressed are those of the authors and do not represent views of the **[Office name]** National Center for Educa or the U.S. Department of Education.

Automated Summary Scoring with ReaderBench

Robert-Mihai Botarleanu[1], Mihai Dascalu[1,2](✉), Laura K. Allen[3],
Scott Andrew Crossley[4], and Danielle S. McNamara[5]

[1] University Politehnica of Bucharest, 313 Splaiul Independentei, 060042 Bucharest, Romania
robert.botarleanu@stud.acs.upb.ro, mihai.dascalu@upb.ro
[2] Academy of Romanian Scientists, Str. Ilfov, Nr. 3, 050044 Bucharest, Romania
[3] University of New Hampshire, Durham, Durham, NH 03824, USA
laura.allen@unh.edu
[4] Department of Applied Linguistics/ESL, Georgia State University, Atlanta, GA 30303, USA
scrossley@gsu.edu
[5] Department of Psychology, Arizona State University, PO Box 871104,
Tempe, AZ 85287, USA
dsmcnama@asu.edu

Abstract. Text summarization is an effective reading comprehension strategy. However, summary evaluation is complex and must account for various factors including the summary and the reference text. This study examines a corpus of approximately 3,000 summaries based on 87 reference texts, with each summary being manually scored on a 4-point Likert scale. Machine learning models leveraging Natural Language Processing (NLP) techniques were trained to predict the extent to which summaries capture the main idea of the target text. The NLP models combined both domain and language independent textual complexity indices from the ReaderBench framework, as well as state-of-the-art language models and deep learning architectures to provide semantic contextualization. The models achieve low errors – normalized MAE ranging from 0.13–0.17 with corresponding R^2 values of up to 0.46. Our approach consistently outperforms baselines that use TF-IDF vectors and linear models, as well as Transfomer-based regression using BERT. These results indicate that NLP algorithms that combine linguistic and semantic indices are accurate and robust, while ensuring generalizability to a wide array of topics.

Keywords: Natural language processing · Text summarization · Automated scoring

1 Introduction

Scoring student writing, which in many cases consists of essays and summaries, is one of the most time-consuming activities teachers have to perform. Yet, it is necessary across the majority of grade levels, academic domains, and in many countries. Teachers must carefully read and evaluate the piece of writing for spelling errors, cohesion and coherence, alignment with the task requirements, plagiarism, and other norms and requirements. Summary evaluation requires even further criteria, such as the faithfulness

A. I. Cristea and C. Troussas (Eds.): ITS 2021, LNCS 12677, pp. 321–332, 2021.
https://doi.org/10.1007/978-3-030-80421-3_35

of the summary to the reference text, the degree to which the summary abbreviates the original reference text, and the objectivity of the summary. The lack of sufficient time for many teachers (who already have excessive burdens) can thus limit opportunities for students to receive sufficient feedback on their summary writing.

In this work, we propose a method for automatically evaluating student summaries to predict main idea coverage. Our aim is to build an Automated Summary Scoring tool that can be used by both students and teachers. For students, the capability to have their summaries evaluated before handing them in would enable an iterative learning process wherein they could write a draft, have it automatically scored, and then work to improve it before the final submission to their teacher. This would allow learners to improve their summary writing skills through a more consistent and timely feedback loop. For teachers, automated scoring can help lower their workload. Automated Summary Scoring systems can support teachers by affording them more time to focus on rhetorical aspects of students' writing, and in turn provide one-to-one assistance to individual students.

One challenge faced by Automated Summary Scoring systems considers their generalization capabilities across topics and target texts. Thus, we address the following research questions:

1. To what extent do summary scoring models generalize across different reference texts?
2. Does performance expressed as Mean Average Error vary when using neural models relying on textual complexity indices or BERT language models?
3. Can novel insights be gleaned about the underlying summary scoring process from feature importance information extracted from the trained neural models?

To achieve these goals, we explored the use of three types of features to predict main idea coverage in summaries: TF-IDF, hand-crafted linguistic and semantic features, and latent contextualized representations computed with BERT [1]. We also examined the efficacy of three types of machine learning models (Random Forest [2], Lasso [3], Neural Networks including feed-forward networks on top of textual complexity indices and BERT). Once trained, we analyze the most important features used by the two best performing models to identify the most relevant information used for automated scoring. In the remainder of this paper, we provide an overview of related work on the automated evaluation of student writing. We then describe our methodological approach and results of our analyses. We then conclude with a discussion of our findings and suggestions for future work.

2 Related Work

There are two primary means through which student writing is automatically assessed [4]: Automated Writing Evaluation (AWE) systems and Automated Essay Scoring (AES) systems. These two systems are commonly used to assess essays, but not summaries. AWE systems offer targeted, constructive feedback to student users with the purpose of helping them improve their writing, whereas AES systems are primarily focused on the generation of a numerical score of writing quality (i.e., a summative score). Here, we present an approach that falls under the category of AES systems.

Various AES systems have been developed to assess multiple genres of writing. e-Rater [5] was one of the first and relies on a wide range of features that measure grammar, usage, mechanics, style, organization, development, lexical complexity, and prompt-specific vocabulary usage. The initial version of e-Rater offered users a method of manually combining these features using weighted averages in an intuitive and explainable system. The past decade has seen considerable progress in the field of text scoring and numerous approaches have been explored. More recent approaches rely on neural networks to score student writing. For example, SkipFlow [6] uses a mechanism for modeling relationships between hidden representation snapshots generated by Long Short-Term Memory Networks [7]. Hochreiter and Schmidhuber [7] trained a network to predict human scores for a set of essays that were written in response to eight prompts. Their model achieved an average Quadratic Weighted Kappa of 0.764, denoting a high level of agreement with the human scores. Alikaniotis, Yannakoudakis and Rei [8] construct a fully automated framework based on LSTMs trained on the same dataset as SkipFlow, with a reported Spearman rank correlation coefficient of 0.91.

Taghipour and Ng [9] used a combination of a convolutional layer to extract local features from the texts, followed by a Recurrent Neural Network to predict the human scores. Similarly, Jin, He, Hui and Sun [10] introduced a two-stage neural network that aims to increase the performance of AES models in prompt-independent contexts. Their network was trained on human-rated essays with different prompts to detect essays with a level of quality that has high deviation from the average; then, these essays were used as pseudo-training data in the second stage.

Our approach consists of a simpler model, based on domain and language independent indices. We also consider the interpretability of our model and attempt to find the most relevant indices used by the Neural Network for our target evaluation criteria.

3 Method

3.1 Corpus

Our corpus consists of 2,976 summaries of 87 reference texts. Expert human raters provided summary scores on seven different analytic measures, which reflect various qualities of the summary and were manually evaluated on a 1 to 4 Likert scale: main idea coverage (“main point”), amount of key conveyed information (“details”), summary cohesiveness (“cohesion”), use of appropriate paraphrasing (“paraphrasing”), use of lexical and syntactic structures beyond those present in the reference text (“language beyond source text”), objectivity of the language used (“objective language”) and summary length. As a proof-of-concept, the current study focuses only on the prediction of the *main idea coverage* criteria. All expert raters were normed on a set of summaries not included in the main dataset. The raters were considered normed once their inter-rater reliability (IRR) reached Kappa .70. After norming, raters scored each summary independently. IRR after independent rating reported Kappa $>$ .60. After independent rating, raters adjudicated any scores that differed by more than one between the raters.

Given the diversity of the corpus, we opted to perform a selection of the test data based on the statistical distributions of the human scores, with the aim of choosing a subset of reference texts and their corresponding scored summaries that provided a wide range of quality. We first combined the seven target scores into a single measure by summing the values for each. We checked for strong multicollinearity (defined as $r >$ .899) and found that none of the variables correlated above that threshold with each other (correlations ranged between .37 and .72). Afterwards, the population variance was measured for each of the 87 reference texts (M = 16.87; SD = 10.10; Min = 1.30; Max = 44.26). Sorting the source texts in decreasing order of their population variance, we then select a number of reference texts that amount to at least 10% of the number of summaries in the corpus and that have at least 30 summarizations.

In developing the test set, we ensured that none of the selected summaries had reference texts present during training and that there was a large number of summaries, with a wide variance of target scores. In the end, our test data was based on three reference texts, included ~10% of the data that included the highest population variance (i.e., the widest range of possible values). This selection guaranteed that the test set contains examples that have both well written summaries, as well as poorly written ones, ensuring that it is sufficiently complex in order to properly evaluate the effectiveness of our models.

3.2 Linguistic and Semantic Features

We used the ReaderBench framework [11] to generate over 730 linguistic and semantic textual complexity indices, covering the following categories:

- **Surface.** Indices that measure statistical attributes of the text such as the number of words, punctuation marks, and character entropies.
- **Morphology.** Indices regarding parts of speech (e.g., noun, verb, adverb).
- **Syntax.** Indices using parse trees to define quantifiable information on the syntactical structure of the text. These include the parse tree imbalance, depth, and others.
- **Cohesion.** Indices derived from Cohesion Network Analysis [12] that measure semantic similarities between text elements (i.e., paragraph, sentences, words).
- **Co-reference.** Indices measuring the length of coreference chains and semantic overlap between words and concepts.
- **Lexical.** Various indices related to lexical features (e.g., hypernymy, polysemy counts, word frequency, word familiarity and lexical complexity.
- **N-gram.** Bi-gram and tri-gram frequencies, such as the number of unique and the total number of n-grams found in a text.
- **Subjectivity.** Frequency of subjective and objective words and phrases.

ReaderBench features were augmented with indices reflecting the degree of overlap between the summary and reference texts, such as their cosine similarities, the Jaccard overlap of their n-grams, and the percentage of the summary that constitutes novel or existing vocabulary with regards to the reference text.

In total, 1466 features were initially considered and were normalized using z-scores. For linear regression models, variance inflation factor was used to filter these features into a subset of 67 that did not exhibit multicollinearity. We also considered applying some common-sense filtering to these indices; for example, using only indices potentially related to text cohesion or summary length targets. However, we found that there are non-trivial interactions between the reference and the summary texts. For instance, reference texts that are already fairly compact in details, but lengthy, would also lead to fairly lengthy summaries. In such a case, even though the absolute number of words in the summary may be larger than in other cases, a summary that manages to preserve key information from the reference text, while performing only minimal shortening, would be found more appropriate than a summary with too much information removed. As such, using indices measured only on the summary text would not give the model enough information to accurately predict the human ratings.

3.3 Machine Learning Models

Machine learning regressors were trained on our dataset to predict the *main idea coverage* score. In order to have baseline evaluations for the selected models and features, we selected to use Random Forest [2] and Lasso Regressors [3]. These models were trained using the ReaderBench linguistic and semantic indices [11] and the vectors representing TF-IDF scores [13]. We also utilize Linear Regressors that use ReaderBench indices to predict the main idea coverage score. Since Linear Regression has issues with multicollinearity that the non-linear models (i.e., Neural Networks, Random Forests) do not, we utilize a Variance Inflation Factor cutoff of 10 [14] to select a subset of indices that does not present multicollinearity. For our models, we elect to not discretize the target score into categorical variables because it has a relevant numerical order, and intermediate values (e.g., a predicted score of 3.2) can still be useful for the user, as they can indicate whether a summary is closer to one range of the rounding interval than the other (e.g., summary is closer to 3.5 than 2.5).

The architecture of the Neural Network model is provided in Fig. 1a. The feed-forward network consists of a single hidden layer alongside ReLu activations [15], together with Batch Normalization layers [16] for controlling covariate shift between layers, and Dropout [17] layers with rate p ranging from 0.2 for the input to 0.5 for intermediate layers (0.5 denotes that half of the inputs are zeroed before being used by the successive layer). This helps control the variance of the model and prevent it from overfitting. The target consists of a single continuous variable for the regressors trained on the main idea coverage score. These models are all trained using a One-Cycle Policy [18] for 50 epochs with a batch size of 8. The optimal learning rate for the One-Cycle Policy was searched in a logarithmic space from 1e−5 to 10 for 70 data points.

We also examined the performance of a BERT [1] model. As shown in in Fig. 1b, the output embeddings were concatenated and then passed through a non-linear layer to perform regression, which was run on both the summary and the reference text. For the BERT model we removed the prediction heads used during pre-training and added a regression head. Since the source texts can exceed the limit of 512 tokens typically used by BERT, we elect to only use the first 512 tokens in these. We have experimented with running the BERT model on blocks of 512 tokens from the source texts and then concatenating the representations; however, the results were poorer than the simpler alternative that trims texts that are too long.

The BERT model was fine-tuned over 7 epochs, utilizing linear schedule with warmup with a learning rate initialized at 0.0001 and the Adam optimizer. We explored different hyperparameter configurations and varied the width and depth of the regression head that uses the BERT outputs and found the best success with using the standard fine-tuning hyperparameters together with a single fully-connected layer used to combine summary and reference features, before estimating the score. Finally, we assessed two models that combined the ReaderBench and BERT indices. Leveraging the architecture illustrated in Figs. 1a and b, the input comprised a concatenation of the document representations generated by BERT, combined with the ReaderBench indices. The combined model attempts to simultaneously finetune BERT and learn to use the ReaderBench indices to predict the target score. The training setup uses the same configuration as the BERT-based model.

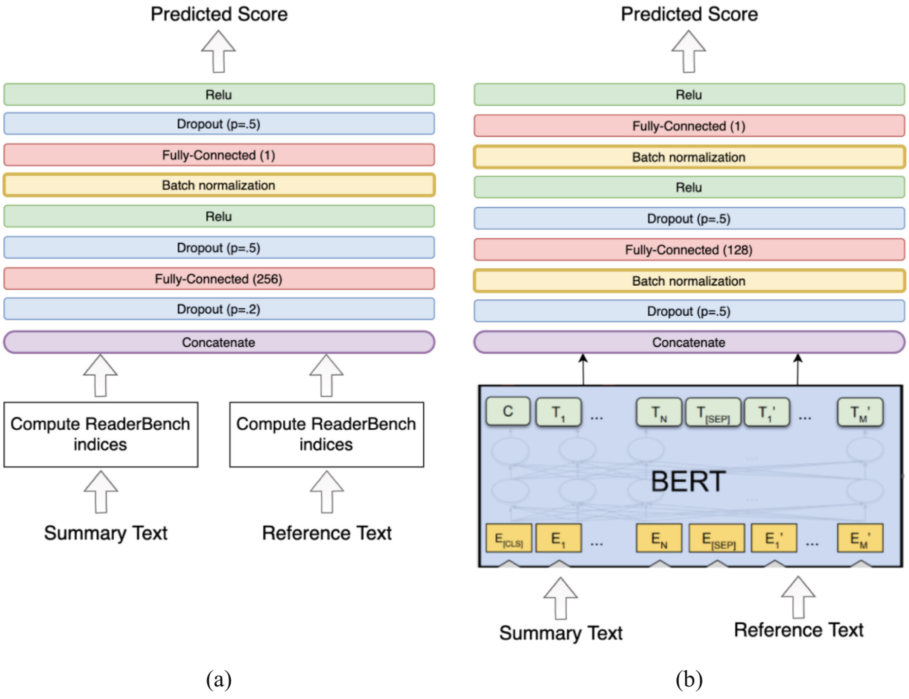

Fig. 1. a. ReaderBench neural network model architecture. b. BERT architecture.

4 Results

4.1 Prediction of Main Idea Coverage Score

The results for predicting main idea coverage scores are presented in Table 1. We can observe that ReaderBench Neural Network model outperforms the BERT and the TF-IDF models. This indicates that the general-purpose language baselines were outperformed, on average, by the networks trained using textual complexity indices. Comparing the three types of machine learning models that used ReaderBench indices, the neural network model tended to yield better results than the other three models.

Table 1. Normalized MAE and R^2 for the "Main Idea Coverage" summarization evaluation criterion.

Models	Normalized MAE	R2
TF-IDF (Lasso)	.17	−.09
TF-IDF (RF)	.17	−.12
ReaderBench: Linear Regression	.16	.15
ReaderBench: Lasso Regression	.17	−.07
ReaderBench: Random Forest	.16	.18
ReaderBench: Neural Network	**.13**	**.46**
BERT	.16	.15
Combined model (ReaderBench & BERT)	.14	.39

4.2 Feature Importance

The relevance of features can be measured for the Random Forest Regressors using the Gini importance, with features being assigned importance values, defined as a normalized measurement of the amount of reduced impurity. Linear models can have their feature importance values directly measured through the feature coefficients after training. Because the neural models used are non-linear networks, we selected Integrated Gradients [19], a method of approximating feature importance by using the gradients resulted from the loss function, for a given sample. Starting from an arbitrary baseline, a line integral is computed along the path from the baseline to the sample, with respect to the feature gradients. This is then scaled with the distance between each intermediate sample and the baseline. The equation describing the integrated gradients of a feature *i*, using a sample x and a baseline x', is the following:

$$IntegratedGradients_{\mathrm{i}}(x) = \left(x_i - x_i'\right) \times \int_{\alpha=0}^{1} \frac{\partial F\left(x' + \alpha \times \left(x - x'\right)\right)}{\partial x_i} d\alpha \quad (1)$$

Integrated gradients, by design, only measure the relative importance of a feature with regards to a given sample and baseline. In our case, the baseline x' is a zero vector (i.e., no indices are measured). The integrated gradients for each feature were measured on the entirety of the training set in order to obtain dataset-wise results, instead of sample-wise results obtained by averaging the sample values.

We present a selection of 5 features from the 10 most important features by the magnitude of their integrated gradients on the best performing model (see Table 2), accompanied by the top 5 features according to their Gini importance for the Random Forest model. For each index, we also specify whether it was measured on the reference (i.e., reference text), the summary, or if it is an overlap index. In addition, we marked each feature with "±" to highlight whether the corresponding gradients have a positive or negative average, before multiplying this value with the difference between the sample and the intermediate baselines. This gives a sense of the directionality of the features. The reason for choosing this method of determining directionality, instead of more traditional approaches (e.g., Spearman rank correlations), is that many important features marked by the model are not linearly correlated with the target variable. The sign of the gradients, on the other hand, should give an indication as to the directionality of the features.

The interpretability of a neural network using integrated gradients is significantly more limited than what the coefficients of a trained linear model can yield. While feature importance is useful as rough guideline, it does not appropriately express the complexity of non-linear interactions that the model uses to make its prediction. Since the model considers more than a thousand features, results in Table 2 give only a shallow understanding of what the model is doing on average, across the testing data. Another important observation is that integrated gradients are commonly used on a per-sample basis, whereas we attempted to extrapolate a global understanding of the behavior of the model, by aggregating the results on each testing sample. The features were chosen to show an equitable distribution of both positive and negative directionalities, in order to give a better insight into the behavior of the model.

Although there is a significant amount of noise in terms of features that have high importance according to the Integrated Gradients method, others are much more intuitive and three of the five are also reported by the Random Forest model. For instance, the presence of overlap features in the main idea coverage score is expected, since this score is dependent on the nature of the original text and how well the summary manages to capture its reference material. Of these overlap features, the "Source-Summary Similarity" is defined as the cosine similarity between the two texts, the "Existing Vocabulary" reflects the vocabulary overlap between them, and the "Jaccard overlap" index measures the similarity between the n-gram sets of the two texts. "Average parse tree imbalance" and the "average block tree depth" are measures of textual structural complexity, while "character entropy" gives a statistical understanding of a text's repetitiveness with low entropy texts typically corresponding to low effort writing. Integrated gradients provide a straightforward measurement of feature importance in neural networks; however, it is a post-hoc interpretation that only approximates the most important features, whereas the non-linearity of neural networks cannot be expressed through simple scores assigned to each input. Nevertheless, the use of integrated gradients and other similar approaches is a way of circumventing the black box nature of modern neural network models and can offer insight into what neural models are actually evaluating during inference.

Table 2. ReaderBench indices with high feature importance used by the NN and RF models

	Neural network	Random forest
1	***Summary – Character Entropy (−)***	*Overlap – Source-Summary Similarity*
2	***Overlap – Source-Summary Similarity (+)***	*Overlap – Source-Summary Existing Vocabulary*
3	Summary – Average Parse Tree Imbalance (−)	Overlap – Source-Summary Jaccard Overlap
4	***Overlap – Source-Summary Existing Vocabulary (+)***	Summary – Average Block Tree Depth
5	Source – Character Entropy (−)	*Summary – Character Entropy*

Note: “ ±” indicates positive or negative gradient values before multiplying this value with the difference between the sample and intermediate baselines. Common indices between the NN and RF models are marked in italics and grayed out cells

5 Conclusions

We performed predictive modelling on a dataset consisting of 2,976 summaries on 87 reference texts to predict main idea coverage. Our results show that, for datasets of this size, the use of hand-crafted features is still very important, with models trained using a variety of textual indices outperforming on average the results of both classic Machine Learning models (such as those based on TF-IDF scores) and state-of-the-art language models (such as BERT). The limitations of BERT, which was designed with larger datasets of shorter texts in mind, made it so that a simpler, fully-connected model, was able to outperform it consistently across different variations and hyper-parameter configurations on our dataset. We relied on a rigorous approach of selecting a testing set such that it precludes any sort of look-ahead bias. In addition, we introduce integrated gradients in the context of using neural networks, together with hand-crafted textual features to better understand what non-linear models are evaluating with regards to the features they learn to use.

Based on our analyses, we found it necessary to use both features that were generated on the reference and the summary text separately, as well as features that were constructed using both texts simultaneously (e.g., the vocabulary overlap). Our feature importance analyses highlighted interesting relationships between the ReaderBench linguistic features and the target variable. Both the Neural Network and the Random Forest models indicate that the semantic similarity between the source text and the summary is an important criterion when scoring the main idea coverage. In addition, the usage of a similar vocabulary to the source text leads to an increase in a summary's score. The evaluations on what the neural model emphasized during inferencing through Integrated Gradients can provide insights into how humans evaluate summaries.

There are several limitations to the proposed method. First of all, the size of the corpus may explain the lower results for the BERT architecture in comparison to the algorithm that uses textual complexity indices because large-scale Deep Learning models, like BERT, benefit from having access to more data during training. Limited datasets, such as the one used in this paper, may often lead to loss of generalization for deep models. Our choice of combining the seven human rating criteria for test set selection offers a proxy towards the holistic view humans develop while evaluating a summary; however, the limited number of data points may have introduced biases. Finally, our method for analyzing dataset-level importance of the different features offers some insight into the mechanisms of the neural network; however, Integrated Gradients is usually used on a sample-by-sample basis. For a certain sample, Integrated Gradients provides an indication as to which sample features are more important, by looking at the gradients that are propagated backward through the network from the loss function. The estimated feature importance is closely tied to the internal mechanisms of the model because the network is updated constantly during training through gradients. Nevertheless, averaging these gradients over the entire dataset can result in certain model behaviors being masked because they are less frequent.

Future avenues of research include the exploration of ways of integrating human domain knowledge to build a model that more closely resembles what humans focus on while evaluating summaries. Our approach considered analyzing the importance of linguistic features after training. The integration of human evaluator preferences into the system could help increase the confidence that tutors have in such systems. One possibility consists of positively weighting features that human evaluators deem most relevant when evaluating a summary, thus encouraging the model to focus on them, while still ensuring the freedom of finding unexpected feature interactions.

With a normalized mean absolute error of 0.13, our results indicate that the best model is capable of matching human evaluations with an average deviation of only 13%. This error rate appears reasonable, if not exceptional. However, the real issue with automated summary scoring systems is whether they are useful to the end-users, namely to students and their teachers. For this, future studies in real-world settings will be necessary to provide a better assessment of the impact of the system. Our approach of performing a post-training analysis in order to identify the features that the model focuses on can help build confidence in the generated scores, through correlating these with human preconceptions. This process can help identify both possible biases in how the scores are assigned, as well as better inform the development of automated summary scoring systems in general through better feature engineering.

Acknowledgments. The work was funded by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number TE 70 *PN-III-P1-1.1-TE-2019-2209,* ATES – "Automated Text Evaluation and Simplification". This research was also supported in part by the Institute of Education Sciences (R305A190063) and the Office of Naval Research (N00014-17-1-2300 and N00014-19-1-2424). The opinions expressed are those of the authors and do not represent views of the IES or ONR.

References

1. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2018). arXiv preprint: arXiv:1810.04805
2. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
3. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. Roy. Stat. Soc. Ser. B (Methodol.) **58**(1), 267–288 (1996)
4. Roscoe, R.D., Varner, L.K., Crossley, S.A., McNamara, D.S.: Developing pedagogically-guided algorithms for intelligent writing feedback. Int. J. Learn. Technol. 25, **8**(4), 362–381 (2013)
5. Attali, Y., Burstein, J.: Automated essay scoring with e-rater V.2.0. In: Annual Meeting of the International Association for Educational Assessment, p. 23. Association for Educational Assessment, Philadelphia (2004)
6. Tay, Y., Phan, M.C., Tuan, L.A., Hui, S.C.: SkipFlow: Incorporating neural coherence features for end-to-end automatic text scoring. In: Thirty-Second AAAI Conference on Artificial Intelligence. AAAI, New Orleans (2018)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
8. Alikaniotis, D., Yannakoudakis, H., Rei, M.: Automatic text scoring using neural networks (2016). arXiv preprint: arXiv:1606.04289
9. Taghipour, K., Ng, H.T.: A neural approach to automated essay scoring. In: EMLP, pp. 1882–1891. ACL, Austin (2016)
10. Jin, C., He, B., Hui, K., Sun, L.: TDNN: a two-stage deep neural network for prompt-independent automated essay scoring. In: 56th Annual Meeting of the ACL Vol. 1: Long Papers, pp. 1088–1097. ACL, Melbourne (2018)

11. Dascalu, M., Dessus, P., Bianco, M., Trausan-Matu, S., Nardy, A.: Mining texts, learner productions and strategies with ReaderBench. In: Peña-Ayala, A. (ed.) Educational Data Mining: Applications and Trends, pp. 345–377. Springer, Cham (2014)
12. Dascalu, M., McNamara, D.S., Trausan-Matu, S., Allen, L.K.: Cohesion network analysis of CSCL participation. Behav. Res. Methods **50**(2), 604–619 (2018)
13. Ramos, J.: Using TF-IDF to determine word relevance in document queries. In: 1st Instructional Conference on Machine Learning, vol. 242, pp. 133–142. ACM, Piscataway (2003)
14. Craney, T.A., Surles, J.G.: Model-dependent variance inflation factor cutoff values. Qual. Eng. **14**(3), 391–403 (2002)
15. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 807–814 (2010)
16. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift (2015). arXiv preprint: arXiv:1502.03167
17. Dahl, G.E., Sainath, T.N., Hinton, G.E.: Improving deep neural networks for LVCSR using rectified linear units and dropout. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 8609–8613. IEEE, Vancouver (2013)
18. Smith, L.N.: A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay (2018). arXiv preprint: arXiv:1803.09820
19. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks (2017). arXiv preprint: arXiv:1703.01365