Improving Automated Scoring of Student Open Responses in Mathematics

Sami Baral
Worcester Polytechnic Institute
sbaral@wpi.edu

Anthony F Botelho
Worcester Polytechnic Institute
abotelho@wpi.edu

John A Erickson
Worcester Polytechnic Institute
jaerickson@wpi.edu

Priyanka Benachamardi
Worcester Polytechnic Institute
pbenachamardi@wpi.edu

Neil T Heffernan
Worcester Polytechnic Institute
nth@wpi.edu

ABSTRACT

Open-ended questions in mathematics are commonly used by teachers to monitor and assess students' deeper conceptual understanding of content. Student answers to these types of questions often exhibit a combination of language, drawn diagrams and tables, and mathematical formulas and expressions that supply teachers with insight into the processes and strategies adopted by students in formulating their responses. While these student responses help to inform teachers on their students' progress and understanding, the amount of variation in these responses can make it difficult and time-consuming for teachers to manually read, assess, and provide feedback to student work. For this reason, there has been a growing body of research in developing AI-powered tools to support teachers in this task. This work seeks to build upon this prior research by introducing a model that is designed to help automate the assessment of student responses to open-ended questions in mathematics through sentence-level semantic representations. We find that this model outperforms previously-published benchmarks across three different metrics. With this model, we conduct an error analysis to examine characteristics of student responses that may be considered to further improve the method.

Keywords

Open responses, Automated scoring, Natural Language Processing, Sentence-BERT, Mathematics

1. INTRODUCTION

In many K-12 mathematics classrooms, teachers have come to rely on the use of open-ended questions to assess their students' knowledge and understanding of assigned content. Unlike close-ended problems, where there is a single or finite-number of accepted answers (e.g. a multiple-choice question), open-ended questions allow students to justify and express their thinking processes through language; it is common that students may combine language, images, tables, or other mathematical expressions, equations, and terminologies to illustrate their knowledge and understanding of the material.

While the use of open-ended questions is not found only in mathematical contexts, aspects of this domain make it particularly difficult to develop teacher supports for these types of question. Within computer-based learning platforms, research across fields of study have led to the development of a multitude of teacher-augmentation tools [1] and methodologies that leverage machine learning techniques. Among these supports, automated methods have been developed and deployed to help teachers assess student essays and short answers in several domains [25, 2, 3, 15]. As was highlighted in [9], the arduous task of manually assessing and providing feedback to student open-ended work may explain the decline of open-ended questions assigned over the course of a school year (e.g. Figure 1 which shows the number of open response questions assigned within the ASSISTments learning platform, aggregated over the last 10 years). In addition to this decline, as was also reported in [9], very few student responses to open-ended questions are ever scored by the teacher, with even fewer ever receiving feedback. Figure 2 illustrates this, as well as the subsequent plot of these values from February through October of 2020, during COVID-19 induced remote learning.

There are several notable challenges in developing automated supports to help teachers assess student open-ended work. It is also the case that student responses to open-ended questions differ in the context of mathematical and non-mathematical domains. One such difference, for example, is that many non-mathematical domains such as history or language arts, student "open-ended" essays and short answers are often comprised of multiple sentences and paragraphs [21, 25, 5, 8], whereas in mathematics, responses are generally shorter (maybe one or two, often incomplete sentences) [14, 9] that combine language with mathematical symbols, expressions, or other visuals. Aside from these response-level characteristics, however, several other student-, problem-, and even teacher-level factors can make the development of

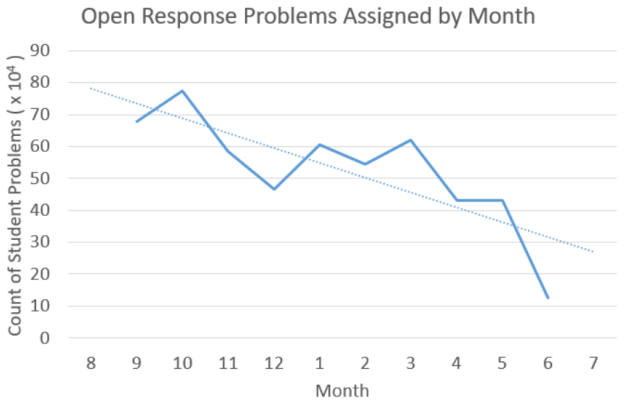

Figure 1: The number of open response problems assigned over the course of a school year with the ASSISTments learning platform, aggregated from 2010-2020.

these automated supports more challenging; consider, for example, the variation in how teachers approach the assessment of student answers, using different inherent rubrics and pedagogical philosophies [15, 17, 22, 23].

While the examination of student answers to open-ended poses challenges in developing automated assessment supports for teachers, prior work has shown promise in this context [9]. In that work, the authors explore several machine-learning and natural language processing (NLP) methods to predict teacher-provided scores to open-ended problems, offering an evaluation method and benchmark of comparison for similar methods[1]

In this paper, we build upon prior research presented in [9] to develop and evaluate an automated assessment model of student open responses in mathematics. We introduce a modeling approach using a sentence-level semantic representation of the student open responses to the existing models through Sentence-BERT (SBERT;[20]), using a novel reformulation of the "score prediction" problem. We compare our method to the previously-developed scoring models from [9], and subsequently apply an exploratory error analysis to identify areas of improvement that may be addressed by future iterations of these methods. Toward this, we seek to address the following research questions:

1. How does a model utilizing Sentence-BERT compare to previously developed approaches in predicting teacher given assessment scores for student response to open-ended problems?

2. What are the characteristics of student answers that correlate with errors observed in our Sentence-BERT model?

3. Which of student-, problem-, or teacher-level characteristics most explain the variance of error observed when the model is applied in real learning environments?

[1]The data and evaluation code from [9] was used in this work with permission from the original authors and in compliance with IRB.

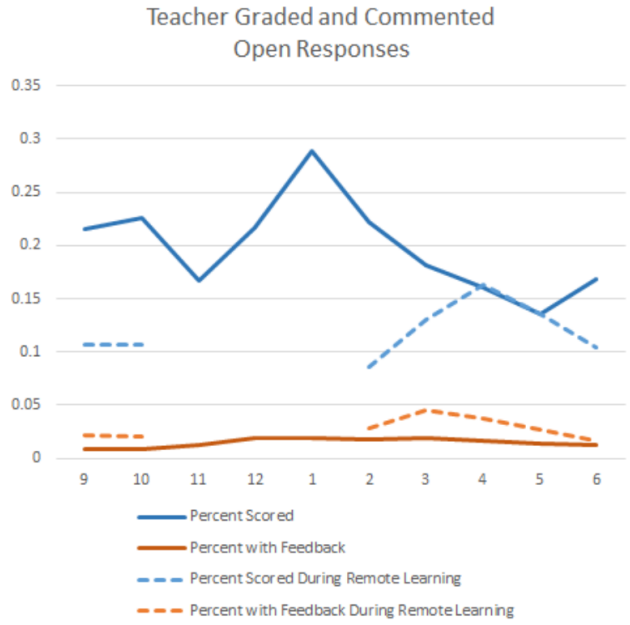

Figure 2: The percent of student open-response answers that were scored and given written feedback by a teacher before and during remote learning in response to COVID-19.

2. BACKGROUND

There have been several works related to the automated scoring of open-ended responses in the past. Most of such works utilize a combination of Natural language Processing (NLP) and machine learning techniques of ranging complexity to process open-ended responses. Much of the existing work in this area has been applied in the context of non-mathematical content. Developments such as C-rater[15] is a well-cited approach that uses such methodologies to estimate the assessed correctness of answers to short answer questions. This method uses grading rubrics and breaks down scores into multiple knowledge components to evaluate each student response. Other works [2, 3] have implemented clustering techniques to grade short textual answers to questions. More recently, studies have based their approach around deep learning methods, which have led to promising improvements over previous benchmarked results [21, 25]. While most of these works have been on non-mathematical domains, studies like [14] explore mathematical language processing using clustering techniques and the bag-of-words approaches for automated assessment of open-ended response in mathematics. However, this study only considers the mathematical content, discarding the non mathematical texts.

Many of these more-recent studies have utilized publicly-released embedding methods trained on large corpuses of data, including those of Word2Vec [18] and GloVe [19], to model the semantic meaning of words. However, word embeddings capture limited information about the semantics of a sentence, where the sequence of words may have large im-

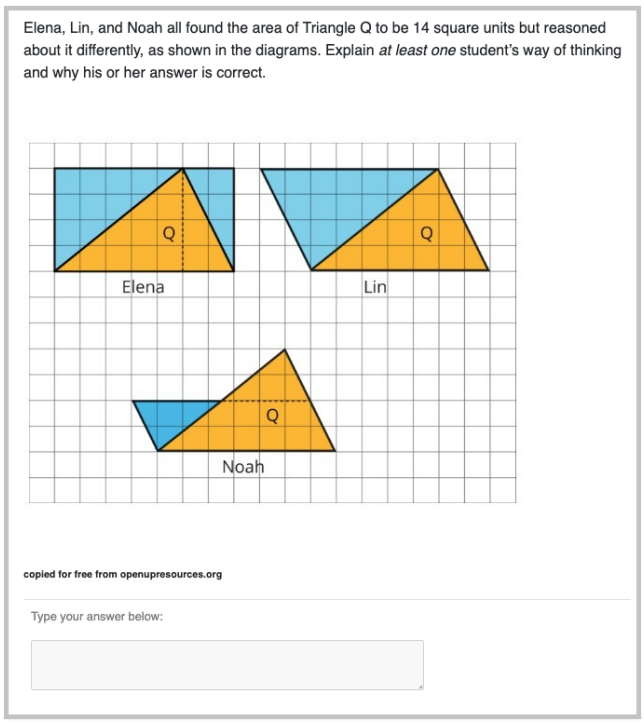

Figure 3: Example of an open ended question taken from openupresources.org

pacts on interpreted meaning. To capture the contextual information within sentences and further increase the generalization capabilities of NLP embedding methods, techniques such as Universal Sentence Encoders [4] and Sentence-BERT [20] generate a single embedding that is designed to be representative of the entire sentence while preserving the semantic and contextual information of the words within such sentences.

One of the most commonly-used NLP embedding methods in recent years has been that of Bidirectional Encoder Representations from Transformers (BERT, [7]). Building upon and distinguishing itself from other methods such as GloVe, the BERT method is designed to incorporate contextual information into generated embeddings to distinguish words that may have the same spelling but different meanings depending on usage (e.g. the word "bank" referring either a financial institution or perhaps a slope of land near a river); BERT has been shown to outperform many other approaches in a number of NLP tasks including, as is important for this work, semantic textual similarity (STS) [7]. Sentence-BERT, or SBERT [20], modifies the pre-trained BERT network to reduce the computational overhead of BERT in order to also generate a sentence-level embedding of a given series of words.

2.1 A Benchmark Comparison

In this work, we are exploring the use of this SBERT method to build upon the prior benchmark set in Erickson et al., 2020 ([9]) in assessing student answers to open-ended problems in mathematics. In that work, the authors discuss the challenges in developing models to predict teacher assigned grades for student open responses in mathematics, using a dataset of authentic student responses within the ASSISTments [11] learning platform. Erickson et al. compares 6 models utilizing machine learning (e.g. random forest and XGBoost [6]) and more complex deep learning (e.g. LSTMs [12]) techniques, combined with natural language processing algorithms to assess responses that are combinations of mathematical expressions and non-mathematical text. For the feature extraction process from the open response data, the study uses the Stanford Tokenizer [16] combined with Global Vectors for Word Representation (GloVe) [19].

3. METHODOLOGY

In this study, we build upon the work of [9] to develop and evaluate an automated scoring model based on the SBERT methodology; as will be detailed further, we refer to this model as the SBERT-Canberra model throughout the remainder of this work. Then, in a secondary analysis, we utilize real data collected from a pilot study of our model running within a computer-based tool that provides teachers with suggested scores to explore the limitations of our approach through an exploratory error analysis. Our data and approach to these analyses are described in this section.

3.1 Dataset

In this work, we utilize two datasets[2] of student answers to open-ended questions paired with teacher-provided assessment scores. An example of one of these open-ended mathematics questions is shown in Figure 3. In this example, students are not asked to find the area of the triangles, but rather explain in their own words what one of the figures is illustrating an approach to solving the problem.

For the development of our SBERT-Canberra model, we use the dataset (and evaluation code) from the Erickson et al. study [9]. This dataset is comprised of student answers to open response questions within the ASSISTments[11] online learning platform; the dataset consists of 150,477 total student responses from 27,199 unique students to 2,076 unique problems graded by 970 unique teachers. As was performed in [9], we omit any case where a student response contained no characters (e.g. an empty response or one containing only whitespace characters), or contained nothing but an image (cases where there was an image accompanied by other text or non-whitespace characters is not omitted). The removal of such empty responses resulted in the dataset dropping to 141,612 graded student responses, 25,069 unique students, 2,042 unique problems, and 891 unique teachers. Within this data, each response is accompanied by a teacher-provided assessment score that follows an integer ordinal 5-point scale from 0-4; a "4" here is synonymous with a student receiving a 100% for the response.

Table 1 lists several student answers contained within the dataset, chosen from across multiple problems for illustrative purposes. As was noted in the introduction, these responses highlight some of the challenges of this modeling

[2]The data and code used in this work cannot be publicly posted due to the potential existence of personally identifiable information contained within student open response answers. In support of open science, this may be sharable through an IRB approval process. Inqueries should be directed to the trailing author of this work.

Table 1: Sample student responses (selected from across multiple problems for illustrative purposes) and the teacher provided scores on a scale of 0 to 4 to the open ended questions in mathematics.

Sample Responses	Score
y=4x-2	4
I counted	4
I multiply -3 and 2x	2
diagram is on paper	3
Yes Because Y=mx+b	0
I got 2/9 by dividing by 4	3
I was not in class for this so I don't know.	1
I went multiplication first then division then multiplication	3
I got this by doing 45/75. I knew that 75 + 75 = 150 and 150 goes into 450 3 times and 3 x 2 = 6. So the answer is 6.	4
You would need an example and then you would need to draw a line and find out far away your shape is from the line and mark it and then do that on the rest of your lines on the shape	4
The distributive property means that a number outside a set of parentheses can be multiplied by each of the numbers within the parentheses and the answer will be the same. It works because it would be the same as multiplying each number by the number outside the parentheses and then adding them together.	1

task. First, the length of responses varies greatly between students as well as across problems. In addition to this, the interleaving of mathematics and linguistic text likely makes it difficult for pre-trained embedding models to interpret. Similarly, the variation in mathematical representation (i.e. the use of the term "dividing" rather than the "/" operator), may lead to confusion in a machine learning model trained over such data. As the mathematical variables are also represented by recognized english characters (e.g. "y"), it may be difficult to derive semantic meaning for such tokens. It is for this reason that we hypothesize that a contextual-based embedding approach, such as BERT and SBERT, may be superior to traditional embedding methods that do not account for context within the sentence. Finally, the noise in ground truth labels become evident from the table. The student who answered "I counted" but still received full credit, for example, exemplifies that some teachers may score students based on completion or other factors unrelated to their demonstration of understanding or mastery. This is not to say that any one scoring method is more correct or valid than another, but rather that there is likely large variation in these labels, making it difficult for machine learning models to effectively learn associations between student answers and these scores in some cases.

The second dataset used in this work is comprised of student responses collected during the pilot testing of a teacher-augmentation tool designed to aid in the assessment of student open response answers within ASSISTments [11]. This tool, called QUICK-Comments, used our developed model to predict the scores of student answers to open response questions in mathematics. Models were trained over the same open educational resource (OER) curricula from which the problems used in the first dataset were collected and produce estimates using the same grading scale as the first study. During the pilot study, 12 middle school mathematics teachers were given access to the tool and compensated for their time to assign, assess, and provide feedback to student open ended work during the Spring and Fall of 2020. This dataset consists of 30,371 graded student open responses to 915 unique open response problems solved by 1,628 unique students.

3.2 SBERT-Canberra Model

The model developed for this work follows a 2-stage process to generate estimates of teacher-assigned scores for a set of given student answers. In approaching this model, we propose a reframing of the initial problem. In [9], the problem was posed as a traditional supervised learning problem; in other words, given a set of student answers A, train a model $f(.)$ such that $Y = f(A)$. Instead, we propose a more unsupervised approach as depicted in Figure 4. If we have a set of historic answers $A_{0...n-1}$, and want to predict the score of a new answer A_n, a logical choice of score may be that corresponding with the historic answer that is most similar to the new answer A_n. In this way, the problem is posed as a similarity ranking problem rather than a supervised learning problem.

There are several potential advantages to this approach. First, when utilizing a pre-trained model of SBERT, described in Section 2, no actual model training is necessary (so long as a reasonable distance metric is identified). Second, as SBERT is optimized for contextual similarity tasks, the problem is better suited to utilize the embedding method's strengths. Finally, in a practical sense, as no model training is necessary (beyond utilizing the pre-trained embedding model), such a model can be more easily applied at scale, requiring just a pool of historic answers to compare against. We hypothesize that this method may also require fewer example answers than traditional machine learning methods as well, but this claim is not deeply explored in this current work.

In applying this method, the set of historic answers $A_{0...n-1}$ are fed through the pre-trained SBERT model to produce

Table 2: Features for the Linear Model of Error analysis of SBERT-Canberra model

Title	Description	Mean
Answer Length	Length of the answer	10.39
Average character per word	the average number of characters per words	3.54
Numbers count	total number of digits	3.54
Operators count	total mathematical symbols in the response	1.47
Equation percent	percentage of mathematical equations in answer	0.27
Presence of Images	Indicator of presence of images in the answer	0.15

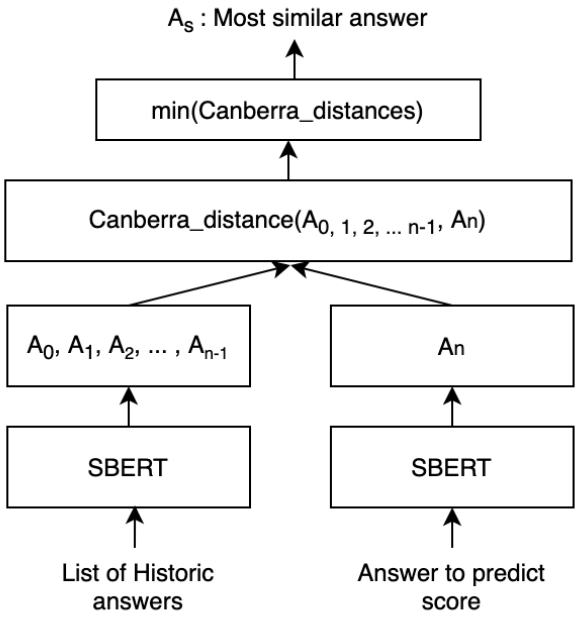

Figure 4: The design of the SBERT-Canberra method, that suggests scores based on similarity between the answers.

a 768-valued feature vector for each answer; these vectors are then stored for later access.. Given a new answer, A_n, a feature vector is similarly produced. In stage two of our method, all pairwise comparisons are then made between A_n and $A_{0...n-1}$, calculating Canberra distance [13] for each pair. Canberra distance, as opposed to other common distance metrics such as Euclidean or Cosine similarity, is a distance metric calculated over ranked lists. With this metric calculated for all pairs, the $A_{0...n-1}$ historic answers are then min-sorted to identify the most similar historic answer, A_s, to our new answer A_n. The score associated with A_s is then used as the prediction for the given answer A_n. The design to this approach is outlined in Figure 4.

As an additional component of this model, a "fallback" condition is implemented to be able to produce scoring estimates for problems where there are no historic answers on which to compare. In this case, we train a single multinomial regression model over all known answers, utilizing 1) the number of words in the answer and 2) the average length of each word in the answer; this model produces a probability distribution over 5 categorical labels (observing the 0-4 grading scale as a multinomial regression formulation). This one model is trained over all known answers and used then only in the case that no historic answers are available for the SBERT-Canberra model. This component is viewed as being part of our SBERT-Canberra approach.

3.3 Evaluation of SBERT

To evaluate our SBERT-Canberra scoring method, we utilize the same data and code presented in [9]. In that paper, the authors present the usage of a 2-parameter rasch model [24] (equivalent to a traditional item response theory, or IRT, model). The purpose of this model is to learn a separate parameter for each student and problem presented, representing student ability and problem difficulty, respectively. The intuition behind the use of this model is to evaluate an NLP automated scoring model based solely on its ability to interpret the words in each student answer. As the score of each answer is likely correlated with student ability (or knowledge) and problem difficulty (e.g. easy problems are likely to exhibit higher scores), such a model provides a reasonable minimum baseline of comparison. By adding a model's scoring estimates as covariates to the rasch model and then comparing the performance of such a model to the rasch model without covariates, we are able to observe the true value-added performance of the NLP scoring model.

Following the same procedure as conducted in [9], we are able to directly compare our Sentence-BERT method to those presented in that prior work. The models are trained and evaluated using a 10-fold student-level cross validation, and model performance is compared based on 3 performance metrics. First, treating the label as multinomial, rather than ordinal, AUC is caluclated using the method described in [10]. Second, the root mean squared error (RMSE), is calucalted over the ordinal prediction and label. Finally, a multi-class kappa is calculated, again using the multinomial label representation. The multinomial representations were argued to be appropriate due to the likely non-linear distribution of scores, while then RMSE provides insight into a more linear assumption of the data. Arguably an additional rank-based metric such as Spearman's Rho would also be a suitable metric of comparison, but is not included for more direct comparisons to the previous work.

3.4 Approach to Error Analysis of the SBERT-Canberra Method

In evaluating the SBERT-Canberra method, it is important to explore limitations of the approach in order to identify where the model does well and where it may yet improve through future iteration. As such, we also conduct an exploratory error analysis of the method using the data

Table 3: Rasch Model Performance compared to the models developed in Erickson et al.[9]

Model	AUC	RMSE	Kappa
Current Paper			
Rasch* + SBERT-Canberra	**0.856**	**0.577**	**0.476**
Erickson et al. 2020			
Baseline Rasch	0.827	0.709	0.370
Rasch + Number of Words	0.829	0.696	0.382
Rasch* + Random Forest	0.850	0.615	0.430
Rasch* + XGBoost	0.832	0.679	0.390
Rasch* + LSTM	0.841	0.637	0.415

*These rasch models also included the number of words.

collected from the QUICK-Comments pilot study. Toward this, we observe two regression models that observe absolute model error as a dependent variable. By exploring characteristics of student answers in the context of this modeling error, we can observe which aspects correlate most with higher prediction error. Similarly, we apply then a multi-level model to observe which of student-, problem-, and teacher-level identifiers most explains any observed modeling error.

3.4.1 Uni-level Linear Model

The uni-level linear model is based on student answer level characteristics. The student answer level characteristics are comprised of a set of six answer-level features extracted from the student open response data. These features are listed in Table 2. In calculating these features, the answer is first tokenized using the Stanford NLP tokenizer[16], dividing each textual answer into smaller tokens. For example, if the response to a particular problem is "I got 2/9 by dividing by 4", a simple tokenizer splits this response text by spaces which would give the list of tokens as: ("I", "got", "2/9", "by", "dividing", "by", "4"). Then from the tokenized data, we separate the tokens consisting of either digits or mathematical symbols. The number of such tokens is divided by the total number of tokens to calculate the equation percentage[3]. The average equation percentage calculated by the procedure mentioned above is 27% across the entire dataset. For calculating the length of the answer text, we count the total words in the text simply by splitting them by space. The average length of answers across the dataset is 10.39. Similarly, within each response, the number of numeric digits (i.e. Numbers count) and number of operator characters (i.e. Operators count) are counted independent of the tokens.

ASSISTments[11] allows students to upload images as part of the response to open-ended questions; this is most commonly a picture taken of work done on paper. The response text in such cases includes the URL of the uploaded image to the system. About 15% of the total responses in the dataset contains images. Some of such responses are entirely images, whereas in others, some text is provided as context. Since these scoring models are not yet designed to support images, we hypothesize that the images' presence contributes significantly to the modeling error.

A simple linear regression model is fit to the pilot study student answers, observing absolute model error as the dependent variable. This value is calculated by simply subtracting the predicted score from the teacher-provided label (as a linear label), and taking the absolute value. In this case, a value of 0 would indicate a correct estimate, while higher values (up to 4) represent greater prediction error; we do not differentiate between under- and over-predicting in this analysis.

3.4.2 Multi-level Linear Model

The uni-level linear model observes features that describe characteristics of the student responses, but as described in Section 3.1, modeling error may not be confined to just characteristics of the responses themselves. It is very likely that modeling error can be attributable to other external factors at the student-, problem-, and teacher-levels.

To explore this possibility, we apply a multi-level linear model observing the student answer characteristics as fixed effects, and student, problem, and teacher identifiers as three separate level-2 random effect variables. As it is the case that the same student may write multiple answers within our data, this structure is similar to that of a repeated-measure analysis.

$$\begin{aligned} abs(\text{model error}) = &\text{Answer Covariates} \\ &+ (1|\text{student identifier}) \\ &+ (1|\text{problem identifier}) \\ &+ (1|\text{teacher identifier}) \end{aligned} \quad (1)$$

Again observing absolute prediction error as the dependent variable, this analysis will be able to answer 1) whether the majority of explainable variance exists at the student-answer level or at a higher level, and 2) which of student-, problem-, and teacher-level identifiers most explains variance in our modeling error (e.g. which of these identifiers is most correlated with the error). The equation, expressed as its R code formulation, is reported as Equation 1.

[3]We acknowledge that this feature is a misnomer as it includes numeric terms, operators, and expressions as well as equations, but chose this feature name for sake of brevity.

Table 4: The resulting model coefficients for the uni-level linear regression model and random and fixed effects of the multi-level linear model of absolute error.

	Uni-level Linear		Multi-level Linear	
	Variance	Std. Dev.	Variance	Std. Dev.
Random Effects				
Student	—	—	0.034	0.185
Problem	—	—	0.313	0.559
Teacher	—	—	0.048	0.851
	B	Std. Error	B	Std. Error
Fixed Effects				
Intercept	0.581***	0.017	0.772***	0.070
Answer Length	-0.008***	0.001	-0.009***	0.001
Avg. Word Length	-0.014***	0.003	-0.013**	0.003
Numbers Count	<0.001	<0.001	<0.001	<0.001
Operators Count	-0.006***	0.001	0.002	0.001
Equation Percent	0.443***	0.018	0.080***	0.022
Presence of Images	2.248***	0.021	1.858***	0.028

*p <0.05 **p<0.01 ***p<0.001

4. RESULTS

4.1 SBERT Model

The results of the SBERT model is compared directly to the results from Erickson et al.[9] as shown in Table 3. As can be seen in that table, the SBERT-Canberra method outperformed the baseline as well as all previous models across all three metrics. While the difference in AUC values between our method and the previous best approach is notably small, the difference in both RMSE and Kappa appears to be comparatively larger. To interpret these two metrics, these results suggest that we should expect teachers to agree with our method's estimates 47% of the time accounting for random chance, and is likely to be wrong by just over half a grade-point on average. This also does suggest, however, that there is still plenty of room for improvement of these models.

What is also worth noting from the results of Erickson et al. [9], is the high performance of the baseline rasch model. This emphasizes the difficulty of this NLP modeling task in that the baseline model is using nothing other than the student and problem identifiers; it is able to seemingly predict teacher-provided scores with an AUC above 0.8 without using any part of the student response; there is only a 0.03 AUC difference between that baseline model and our current proposed method. This suggests that these external factors may be explaining a large portion of the student scores, and may subsequently explain a large portion of our prediction error.

4.2 Error Analysis of SBERT

In exploring this further, the results of the error analysis of the SBERT-Canberra method are presented in Table 4. It is found that the uni-level linear model explains 38.6% of the variance of the outcome as given by r-squared. Out of the six student answer-level features, nearly all were found to be statistically reliable predictors of model error; in verifying these results, it was found that all included covariates exhibited inter-correlations less than 0.3 (suggesting a moderately low impact of multicollinearity potentially skewing the interpretation of these results). In close examination of the coefficients of these features, however, despite being statistically reliable, many are found to be close to 0, suggesting a very little meaningful correlation with the modeling error. This is not the case, however, for two of these variables, Equation Percent and Presence of Images, we see a more meaningful coefficient. This suggests, due to the direction of this value, that the presence of mathematical elements as well as the presence of images (unsurprisingly) both correlate with higher prediction error. It further follows, then, that further improvements to the SBERT-Canberra method should explore methods of better representing and accounting for these mathematical terms in student responses; similarly, though likely much more difficult, incorporating an aspect of image recognition could be another area worth exploring.

In regard to the multi-level linear model, accounting for student, problem, and teacher identifiers each as random effects, we see that the inclusion of these level-2 factors explains some of the impact of the fixed effects (also in Table 4). Here it is found that all but two of the fixed effects are statistically reliable. It is also found that the magnitude of the coefficients for the Equation Percent and Presence of Images is also reduced. This suggests that, perhaps, the student and/or problem identifiers partially explain these correlations (some problems may be more likely to have responses with images or mathematical terms in them, or some students may be more inclined to use images or such terms more than others). What is worth noting, however, is that it was found that the level-2 variables account for 55.5% of the variance of the outcome. This suggests that a majority of the modeling error can be explained by these factors that are external to the student answers.

Looking at the variance of the random effects, it can be seen that the problem level identifiers contribute most in terms of explaining the variance of the outcome. It is certainly the

case that the SBERT-Canberra method is accounting for each individual problem in producing its estimates (e.g. it only observes historic answers within each unique problem), but it would seem that there are other problem-level factors that are not being accounted for within this approach.

5. LIMITATIONS AND FUTURE WORK

In regard to our approach as well as in light of our findings, there are several limitations and opportunities for future directions. While the SBERT-Canberra approach, utilizing sentence-level embeddings, outperforms the previously-developed models in predicting scores for open responses, the difference in AUC is rather small; the fact that the method produces a classification (as opposed to a probability as is often the case with such models) likely impacts its AUC performance. The manner in which the method makes its prediction can be considered a greedy approach in that only the closest historic answer is used to predict the score. Instead, a weighted vote approach using all historic scores (or a subset of similar scores above an identified threshold) may improve the model by allowing for some degree of uncertainty. Similarly, the use of the word count model as a fallback may further be improved; while it was the case that there were very few instances of problems not having enough data within the cross validation, improving this fallback method may help to improve the model when applied in practical settings where the "cold start" problem is more prevalent; as the method currently relies heavily on having a sufficiently-sized pool of human-scored historic answers, future research can focus on utilizing unlabeled student answers or exploring other unsupervised methods that may additionally support these methods in cases where labeled data is scarce.

While the SBERT-Canberra model performed arguably well, the error analysis revealed several areas where this approach, as well as others, may focus in future works. Most notably, as highlighted, the use of mathematical expressions and terms were found to be correlated with higher error; improving the representation of such elements can certainly be addressed in future work. A limitation of this, however, is that both models left variance unexplained in the outcome. We chose to look at these factors based on hypotheses and anecdotal observations, but there may be other large factors that can explain more of the error that we are seeing. Subsequent works could conduct more thorough surveys of both answer-level and higher-level factors. Future works can also explore additional model structures and language features that may lead to improvements to performance. The analyses presented in this work, however, can act as a baseline to further evaluate if future iterations of our approach truly improve upon these identified areas.

It is also the case that this work focuses only on models that predict numeric assessment scores, while we strongly believe that it will be equally, if not more important to additionally develop methods to suggest or generate directed feedback for for these student answers; teachers use textual feedback messages to offer constructive guidance to students, but it is often a very time-consuming task to write these messages for each students' answer. We believe that the SBERT-Canberra approach can be extended to support this task as well, where such a model may be able to recommend feedback to new student answers that has been previously given to an identified similar historic answer. Future work is intended to explore these methods further for such feedback-suggestion tasks.

6. CONCLUSION

In this paper, we have presented a novel approach in addressing and formulation of the problem of automating the assessment of student open-ended work. We have illustrated that our SBERT-Canberra method outperformed a previously-established benchmark, but still exhibits areas where it may be able to improve. Through the conducted error analysis, we have identified areas where more advanced methods of image processing and natural language processing (or math language processing), may lead to further improvements. With all of this, however, it was also identified that problem-level features appear to be most impactful in explaining the variance of modeling error; this is particularly surprising as variations in teacher grading were previously hypothesized to be a larger factor in this context.

With the findings from the study, our goal next is to use them to overcome the limitations mentioned above and guide our focus on improving the methods for assessment of open-ended questions in mathematics. It is the goal of this work to act as a step toward building better teacher supports for these types of open-ended problems, as well as provide others with guidance toward the same or similar goals.

7. ACKNOWLEDGMENTS

We thank multiple NSF grants (e.g., 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, 1535428, 1440753, 1316736, 1252297, 1109483, & DRL-1031398), as well as the US Department of Education for three different funding lines; the Institute for Education Sciences (e.g., IES R305A170137, R305A170243, R305A180401, R305A120125, R305A180401, & R305C100024), the Graduate Assistance in Areas of National Need program (e.g., P200A180088 & P200A150306), and the EIR. We also thank the Office of Naval Research (N00014-18-1-2768) and finally Schmidt Futures we well as a second anonymous philanthropy.

8. REFERENCES

[1] P. An, K. Holstein, B. d'Anjou, B. Eggen, and S. Bakker. The ta framework: Designing real-time teaching augmentation for k-12 classrooms. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2020.

[2] S. Basu, C. Jacobs, and L. Vanderwende. Powergrading: a clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics*, 1:391–402, 2013.

[3] M. Brooks, S. Basu, C. Jacobs, and L. Vanderwende. Divide and correct: Using clusters to grade short answers at scale. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 89–98, 2014.

[4] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Céspedes,

S. Yuan, C. Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[5] H. Chen and B. He. Automated essay scoring by maximizing human-machine agreement. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1741–1752, 2013.

[6] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] S. Dikli. An overview of automated scoring of essays. *The Journal of Technology, Learning and Assessment*, 5(1), 2006.

[9] J. A. Erickson, A. F. Botelho, S. McAteer, A. Varatharaj, and N. T. Heffernan. The automated grading of student open responses in mathematics. In *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, pages 615–624, 2020.

[10] D. J. Hand and R. J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine learning*, 45(2):171–186, 2001.

[11] N. T. Heffernan and C. L. Heffernan. The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497, 2014.

[12] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[13] G. Jurman, S. Riccadonna, R. Visintainer, and C. Furlanello. Canberra distance on ranked lists. In *Proceedings of advances in ranking NIPS 09 workshop*, pages 22–27. Citeseer, 2009.

[14] A. S. Lan, D. Vats, A. E. Waters, and R. G. Baraniuk. Mathematical language processing: Automatic grading and feedback for open response mathematical questions. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pages 167–176, 2015.

[15] C. Leacock and M. Chodorow. C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 37(4):389–405, 2003.

[16] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[17] S. L. Meier, B. S. Rich, and J. Cady. Teachers' use of rubrics to score non-traditional tasks: Factors related to discrepancies in scoring. *Assessment in Education*, 13(01):69–95, 2006.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[19] J. Pennington, R. Socher, and C. Manning. Global vectors for word representation. 2015.

[20] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[21] B. Riordan, A. Horbach, A. Cahill, T. Zesch, and C. Lee. Investigating neural architectures for short answer scoring. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 159–168, 2017.

[22] J. Z. Sukkarieh, S. G. Pulman, and N. Raikes. Automarking: using computational linguistics to score short ,free- text responses. 2003.

[23] D. R. Thompson and S. L. Senk. Implementing the assessment standards for school mathematics: Using rubrics in high school mathematics courses. *The Mathematics Teacher*, 91(9):786–793, 1998.

[24] B. D. Wright. Solving measurement problems with the rasch model. *Journal of educational measurement*, pages 97–116, 1977.

[25] S. Zhao, Y. Zhang, X. Xiong, A. Botelho, and N. Heffernan. A memory-augmented neural model for automated grading. In *Proceedings of the fourth (2017) ACM conference on learning@ scale*, pages 189–192, 2017.