The Automated Grading of Student Open Responses in Mathematics

John A. Erickson
Worcester Polytechnic Institute
Worcester, MA, United States
jaerickson@wpi.edu

Anthony F. Botelho
Worcester Polytechnic Institute
Worcester, MA, United States
abotelho@wpi.edu

Steven McAteer
Worcester Polytechnic Institute
Worcester, MA, United States
smcateer@wpi.edu

Ashvini Varatharaj
Worcester Polytechnic Institute
Worcester, MA, United States
avaratharaj@wpi.edu

Neil T. Heffernan
Worcester Polytechnic Institute
Worcester, MA, United States
nth@wpi.edu

ABSTRACT

The use of computer-based systems in classrooms has provided teachers with new opportunities in delivering content to students, supplementing instruction, and assessing student knowledge and comprehension. Among the largest benefits of these systems is their ability to provide students with feedback on their work and also report student performance and progress to their teacher. While computer-based systems can automatically assess student answers to a range of question types, a limitation faced by many systems is in regard to open-ended problems. Many systems are either unable to provide support for open-ended problems, relying on the teacher to grade them manually, or avoid such question types entirely. Due to recent advancements in natural language processing methods, the automation of essay grading has made notable strides. However, much of this research has pertained to domains outside of mathematics, where the use of open-ended problems can be used by teachers to assess students' understanding of mathematical concepts beyond what is possible on other types of problems. This research explores the viability and challenges of developing automated graders of open-ended student responses in mathematics. We further explore how the scale of available data impacts model performance. Focusing on content delivered through the ASSISTments online learning platform, we present a set of analyses pertaining to the development and evaluation of models to predict teacher-assigned grades for student open responses.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; Machine learning approaches.

KEYWORDS

open responses, automatic grading, natural language processing

LAK '20, March 23–27, 2020, Frankfurt, Germany

ACM ISBN 978-1-4503-7712-6/20/03…$15.00
https://doi.org/10.1145/3375462.3375523

ACM Reference Format:
John A. Erickson, Anthony F. Botelho, Steven McAteer, Ashvini Varatharaj, and Neil T. Heffernan. 2020. The Automated Grading of Student Open Responses in Mathematics. In *Proceedings of the 10th International Conference on Learning Analytics and Knowledge (LAK '20), March 23–27, 2020, Frankfurt, Germany*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3375462.3375523

1 INTRODUCTION

With classrooms progressively adopting free online educational resources (OER's) and curricula, such as Engage New York (EngageNY), Illustrative Mathematics, or Utah Math, a large number of teachers and students are gaining access to expert-authored content. The benefit of using such resources extends to give teachers the ability to assign content aligned with developed standards, supplying them with a range of problems which can be used to provide students opportunities to practice each skill and also can help to assess students' knowledge and understanding of such skills. While the resources themselves provide promise to help teachers gain these benefits, OER's are merely content-based and are not a technology aimed at helping teachers beyond providing the problems and suggested structure of the curriculum.

Conversely, one of the goals of computer-based learning platforms is to help teachers deliver content to students in order to supplement instruction, provide aid to students, and report student learning progress and assessment to the teacher. In doing so, these systems often record large amounts of fine-grained student data to help the teacher make more data-driven decisions in the classroom (e.g. helping to identify which homework problems on which to focus a class discussion). In many cases, this is accomplished through the system's ability to automatically grade student content in order to then report that information back to the teacher.

As open educational resources such as EngageNY, Illustrative Mathematics and Utah Math are more content-focused, and computer-based learning platforms are more instruction-, assessment-, and feedback-focused, the incorporation of OER content into these systems can wed the benefits of each to support both teachers and students. ASSISTments, the learning platform from which we have acquired the data used in this work, is one such system that has incorporated such content. While these learning platforms have many strengths, a current limitation exists in many systems regarding the support for open response questions which comprises a

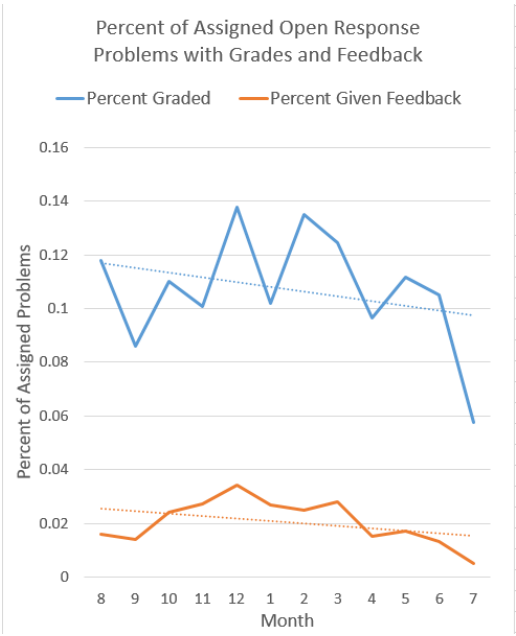

Figure 1: Percent of Assigned Open Response Problems with Grades and Feedback

large percentage of the content within these OERs, but of course open-ended problems are not limited to these content sources alone.

The task of automating the grading of student responses to problems in computer-based learning platforms has largely been limited to well-defined or well-structured types of problems. These types of questions include, for example, those problems which have a standard correct response, such as solving a simple mathematical expression (i.e. 6 * 6 = ?). Likewise, there are those problems which could be represented by a mathematical expression (i.e. solve for x: 8x + 3 = 7) where the correct answer could take the form of a fraction or a decimal value (i.e. 0.5 or 1/2), where either would be considered correct. While these problem types aim to evaluate students' knowledge of a given topic, questions that require students to explain their reasoning further provide the opportunity to assess students' understanding of the assigned concepts. In order to do so, however, the grader needs to be able to parse and, to some degree, understand the semantics of each response to measure the student's comprehension of the material.

Many of the widely used Intelligent Tutoring Systems, such as McGraw Hill's ALEKS^{TM} and Carnegie Learning's Cognitive Tutor^{TM}, have no concept of open response questions, likely due to their inability to automatically assess students' responses. Others, such as ASSISTments, do provide a tool for teachers to grade student responses but make no attempt to automatically grade them. While a wide range of automatic short answer grading systems have been developed and documented [1], grading responses to open-ended questions in mathematics remains a task that teachers predominantly do manually.

In ASSISTments, the manual grading of open responses is not common, likely due to the arduous nature of reading and assessing student work. Figure 1 illustrates the percentage of open response problems in ASSISTments that are ultimately graded by teachers as well as the percentage of such problems where the teacher has provided feedback (i.e. in the form of a comment or message) for the work. In that figure, it is apparent that less than 15% of assigned open response problems in the system are given a grade by the teacher and even fewer (less than 4%) receive feedback. Furthermore, this percentage decreases over the course of the school year, presumably as teachers realize how much time it takes to properly attend to student responses. This figure illustrates the need to provide teachers with better support in assessing student work.

In this paper, we study the viability and challenges of developing models for the automatic grading of mathematics open response questions using data collected from real teachers assigning content within the ASSISTments online learning platform. Toward this goal, we seek to:

(1) Examine variations in teacher grading policies of open responses within ASSISTments
(2) Evaluate how well models are able to predict teacher-assigned grades of student open responses given the currently available data from within ASSISTments.
(3) Investigate how the performance of our models are affected by the scale of available training data.

The goal of this research is to serve as tool and framework for future studies and experiments involving the automated grading of and generation of feedback for open response questions in computer-based learning platforms

a2

2 BACKGROUND

There have been many previous works utilizing natural language processing (NLP) to provide feedback on responses to open-ended short answer essay questions; the specific NLP techniques used in these works, however, have ranged in complexity in an attempt to extract information from the language. Studies such as [22] have developed systems which use hand-crafted pattern matching to grade one-to-two sentence student responses to open-ended questions. Others, like c-rater [21], make use of grading rubrics breaking down scores into multiple knowledge components for evaluation; student responses are parsed to detect the presence of either a paraphrasing of a concept or statements that infer a concept pertaining to such knowledge components. Recent studies like [16] have also shown promising results using neural network models with no need for feature engineering. In many recent works, several deep learning methods, such as Word2Vec [13] and GloVe [14], have been used for their ability to capture the semantic and contextual information of words, while another approach has attempted to use memory-augmented networks to better incorporate labeled examples of essays [25].

While these deep learning methods have gained popularity for use in NLP tasks, the methods often require large amounts of language data to train and pre-trained models may be limited to words that were in the original corpus (which often excludes the specific math words and symbols that may be found in student responses to open-ended questions). It is for this reason that another, albeit

much simpler technique, known as "bag of words" has been applied with some success in certain NLP tasks; this method observes the frequency of each word within and across the given samples, generating a weight measure representing the prominence of that word. While bag of words is a simplistic approach, it is one that has been around for a long time with studies such as [8]. Today, bag of words is the foundation of many studies and strategies. Studies such as Alessandro Sordoni's dynamic context generative models utilize bag of words as an input to their RLMT generating responses from text [20]. In addition, one of the more common approaches, latent semantic analysis, is based on the bag of words approach, essentially allowing for the comparison of the K-dimensional vectors of two bag of words representations and evaluating the match between these vectors [4].

While most of the discussed non-deep learning approaches utilize bag of words, a known flaw is that the structure of the sentence is not understood. A simple approach is a n_gram model. This will allow the model to save and understand spatial information of the sentences. Studies such as [17] utilized this approach to create the variables within their logistic regression to predict course completion.

Although the majority of research pertaining to the automatic grading of student open-ended responses has largely focused on non-mathematical content, there have been several works applied within the domain of mathematics. [10] have explored mathematical language processing for open responses by utilizing clustering methods and bag of words. However, in the case of that study, the focus was on limiting the model to analyze only the mathematical expressions while disregarding text; when the independent variables (i.e. the corresponding prominence of each word) were generated for the model, all non-algebraic text was omitted. This current work, which will be discussed further in the Methodology Section, uses multiple of these approaches including bag of words to include both mathematical expressions and non-algebraic text, and utilizing pre-trained word embedding within deep learning methods to help find the semantics within the open response text.

As it will become more apparent by the description of our data in the Dataset Section, there are several factors that differ between the task described in this work and that of previous related works. These factors can be summarized in terms of the domain of focus, the scale of available data, and the consistency of grading (outcome label). The work of [16], for example, used datasets from state-level assessments spanning science, biology and ELA (English Language Arts), with an average of 2200 responses for each question; in addition to this, the consistency of labels within the data were arguably more consistent as they were scored by two human annotators. Other studies such as [3] and [23], consist of equally large datasets of 80 questions and 2273 student responses (approximately 28 responses for each question) from ten assignments consisting of four-to-seven questions each, and two exams containing ten questions each. Similar to the Riordan study, two human judges were used to score the student responses. In the case of [10], the dataset consisted of 116 learners solving 4 open response mathematical questions (2 high school level math questions and 2 college level signal processing questions) in an edX course. Similarly, [17] utilized a single education focused HarvardX course to collect data from 41,946 enrolled students.

This study aims to enable the ability to automatically grade student open responses within online tutoring systems. As discussed prior, support for open response questions on current platforms, such as ASSISTments, are limited and automatic grading is lacking. As shown earlier, the lack of automatic grading leads to a sharp decline in open response questions as the year gets busier and the efficiency of multiple choice questions becomes more enticing. Studies such as [19] support this, discussing how multiple choice are prioritized for the ease, accuracy and speed of grading. [9] highlighted the advantages to a wider variety of question types; that providing evaluations of just one question type is insufficient in testing the students true critical thinking and understanding. Other studies [12] discussed how constructed response questions (open response questions) elicit a larger range of cognition's than that of just multiple choice.

In the case of standard essay grading (e.g. pertaining to non-mathematics content and ranging from one sentence to multi-paragraph), there are often very large datasets on which to train NLP models as it is understood that large scale is often necessary; the ASAP Kaggle competition [15] is an example of such data that has been made publicly available. Open ended responses in the context of mathematics, however, differs greatly from those observed in other domains as the structure of the language is often secondary to the students' ability to demonstrate knowledge and understanding of the concepts in regard to what is considered when determining appropriate scores. The lack of publicly available data on which to build automated graders of student open-responses, within the domain of mathematics, further makes it difficult for the field to progress in this task. It is for this reason that we not only focus on teacher generated content, but also on OER content. As it is widely used by teachers, supporting an opportunity to make meaningful strides to support teachers on material already being used in real classrooms. While many previous works used a larger pool of data per question or better consistency across labels, our dataset is comprised of student responses to content assigned in true classroom settings by teachers, and is therefore representative of the type of information that would be available to models deployed in such settings.

3 PILOT STUDY: VARIATIONS AMONGST GRADERS

Among the largest challenges in developing models to automate the assessment of student open responses is the subjective nature of grading labels. In systems that allow teachers to manually grade responses to open-ended problems, such as in ASSISTments, such teachers are not prescribed a rubric to follow or a set of criteria by which they must assess students; teachers grade their own students based on how well they feel the student has met their own requirements. In most cases, teachers presumably assess students based on how well they are able to articulate and demonstrate their knowledge of assigned content. Others, however, may also grade based on effort, perhaps based on grammar, or even based solely on completeness rather than the content of the response. While some of these cases can be detected (i.e. teachers who only grade based on completion of the problem), other causes of variation are

likely more difficult to detect and normalize to help a model learn to assess students on a common scale.

In order to better understand the degree to which teachers' grading policies vary, we conducted a pilot study with 14 teachers[1] who use ASSISTments to regularly assign content from open educational resources. As it is normally difficult to measure variations in grading due to differences in both assigned content and the wording of student responses, we presented these teachers with a subset of a group of 125 student responses to a set of 3 problems that had been assigned in the previous month; each teacher was given a random subset of 25 responses from their own and other teachers' students plus an additional set of up to 10 anonymized responses from their own students (e.g. if the random set of 25 student responses contained 5 responses from a teacher's own students, an additional 5 responses from their class were selected for that teacher). This selection process allowed for multiple teachers to grade a same subset of student responses as well as re-grade a subset of their own student responses in an anonymized manner (as the pool of responses were selected from those that had been assigned and graded by the 14 teachers previously).

From this data, we were able to calculate inter-rater agreement on the set of teachers to understand how much variation existed in how teachers assess student open-ended work. We apply Fleiss' Kappa as we have more than two raters per response and found that there was just under 17% agreement above random chance on grades between the teachers (kappa=0.167) when assessing on the 5-point scale. When the grades are dichotomized into a binary value (where a grade less than 2 is treated as a 0 and grades equal to or greater than 2 are treated as 1), the agreement rises to 41% above random chance (kappa=0.417). These levels of agreement are surprisingly low, suggesting that there is very large variation in how teachers approach grading these open responses questions. It was also found, when looking at the internal consistency of teachers' grades of their own students, their Cohen's kappa ranged from 23% agreement (kappa=0.231) to over 67% agreement above chance (kappa=0.677); again, these later kappa values are calculated by observing the agreement between the given grades with those previously given for the responses of their own students (all presented anonymously). In following interviews with these teachers, it was suggested that this low internal consistency may be attributable to other contextual factors that are considered when grading students.

The large variation in grades and potential contextual factors that may exist external to the content of a given open response highlight potential challenges faced in developing models that seek to automate this process; such models need to be able to generalize across teachers and students, and the results of our pilot study suggest that this will be difficult. It is for this reason that we include a teacher-level factor in our analysis described in Section 5.4 and discussed further in the Results Section.

4 DATASET

For the goal of developing models to automatically assess student open responses in mathematics, we collected a dataset comprised of authentic student answers to open response questions within

[1]The teachers in this pilot study were recruited through a funded NSF grant (Blinded for Review)

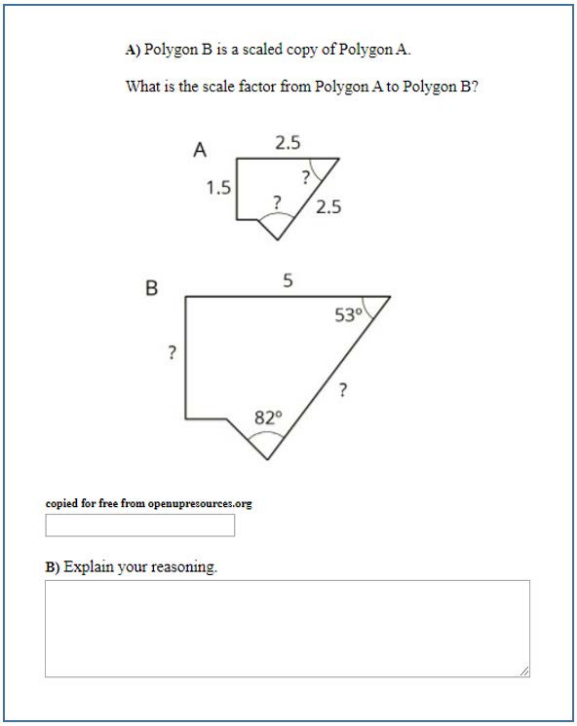

Figure 2: Example Problem Selected from Illustrative Math open educational resource

ASSISTments[6] [18]; while the source of content does vary, a large portion of the open response problems contained within the dataset are from open educational resources such as EngageNY, Illustrative Mathematics, and Utah Math. ASSISTments is used by real teachers and students for classwork and homework, and is developed around the idea of providing immediate feedback to students (on all but open-ended problems) and the reporting of student performance for teachers. As stated in the Introduction Section, ASSISTments has incorporated several OER curricula into its available content, providing the means to collect the student responses to EngageNY, Illustrative Mathematics and Utah Math open-ended questions (as well as others) as they were assigned and graded by teachers.

In the raw and unfiltered state, the dataset consisted of 27,199 unique students with 150,447 total student responses to 2,076 unique problems, and graded by 970 unique teachers. In the data, there were a number of empty responses provided by students caused by either a student submitting nothing (i.e. submitting an answer consisting of only a 'space' character) or by a student submitting an image as their response; images were not included in the data resulting in what appears to be an empty response. As such, any empty responses are omitted from the dataset for the analyses described in subsequent sections, as it is also the case that few would argue that a truly empty response should be given a grade of 0, and the omission of such cases will avoid inflating model performance. Once the filter was applied, the total number of graded student responses dropped to 141,612, the number of unique problems was decreased to 2,042, and left 25,069 unique students and 891 unique teachers.

An example of the types of responses and their variations can be seen in Table 1. What is clear is that there are a wider variety of responses from students, with inconsistent spelling, mathematical functions written differently and random text. This is one of the main challenges of this study. Another challenge presented in the dataset, and of this study, is that each student response is graded by one teacher with the exception of a small number of samples where multiple teachers assessed the same student responses.

4.1 Response Feature Extraction

To support our model development, we take two steps to extract features from the text of the student responses: we first tokenize the student responses and then create a numeric representation of these parsed words using one of several methods. It is common in natural language processing approaches to tokenize, or identify individual words from provided text. For instance, a student may respond with "I didn't know the answer, so I guessed 4" where the text would be divided into each component; a simple approach here would be to simply split the text using spaces, but other approaches may attempt to additionally separate punctuation or contractions into separate components. Within this analysis, the text is tokenized utilizing two different approaches: what is known as standard count vectorizer splitting, as well as the Stanford Tokenizer[11]. This later tokenizer was applied to better support our deep learning approach described later in this section.

To describe the standard count vectorizer, this approach will take our full corpus of responses, split the words and create a list of those words. Table 2 shows an example of the initial processing to extract words/features from student responses. From there, the text which is being trained on is passed through this list, creating a RxW matrix where R is the number of responses by students in the training set, and W is the number of unique words within the overall corpus. Then in each column of W, a count of the occurrence of the word in the student's response is tallied.

In the end, the final RxW matrix acts as the bag-of-words approach described in Section 2; by adding the frequency values, a numeric representation is given to each word describing its weight amongst other words in the corpus. While this representation approach could result in undesirable omissions, where, for example, the method may partition an equation contained in a student response (as shown in Table 2 with *6x4 = 6* recognized as simply *6x4*); it does, however, allow for more flexibility in capturing similar numeric occurrences. As is the case in a bag-of-words approach, the ordering of words within each response is not maintained by the representation and instead relies on a measure of word prominence. With just the count, it is apparent that certain 'stop' words such as 'i', 'me', 'my', 'it', 'this', 'that' would carry more weight given that the words are used often. To combat this, the term frequency-inverse document frequency (tf-idf) statistic is calculated across the matrix. These features will later be used in the non-deep learning models described in the next section.

The other approach to extracting the features from text utilized in this study is the Stanford Tokenizer [11] combined with Global Vectors for Word Representation (GloVe)[14]; GloVe word embeddings, pre-trained on large datasets, have been made openly available to researchers conducting natural language processing research. In the pre-trained embeddings used in this work, the Stanford Tokenizer was used in the generation of such word representations, so the same tokenizer is applied to maximize the number of words recognized by that model. The Stanford Tokenizer was applied, which increases the amount of words which are able to be respresented by a GloVe vector. For example, the Stanford Tokenizer will represent "didn't" as "did" and "n't" which is necessary for the pre-trained GloVe model to recognize each component (i.e. there is no pre-trained GloVe representation for "didn't" but there are representations for "did" and "n't"). We used the 100-dimensional GloVe vectors pre-trained on a large Wikipedia dataset with the hypothesis that such a corpus is more likely to include mathematics terms than other pre-trained models using, for example, news sources.

5 METHODOLOGY

To develop our models, we use both traditional machine learning techniques and more complex deep learning algorithms combined with natural language processing approaches. The range of models observed in this work is intended to compare models of varying complexity and flexibility in regard to how such models represent the presented data. Specifically, we compare two decision tree-based models with a deep learning network within the context of a probabilistic baseline model.

With the tree-based machine learning approaches including random forest and XGBoost, each described within this section, there is a decrease in flexibility of the model (in comparison to deep learning algorithm's such as a neural network or LSTM), but greater likelihood in being able to interpret results and identify impactful words/equations within the student's response in order to justify the model's prediction (a potentially desirable quality of a model that will be suggesting grades to teachers). With the inclusion of deep learning, our analysis is taking advantage of the newest approaches and allows us utilize embedding's to help our models understand the semantics of the words/equations within the student's response. Additionally, the final models utilize the predictions from the traditional machine learning and deep learning models as covariates within a Rasch model. The following sections detail the methodologies applied to address the goals outlined at the end of the Introduction Section.

5.1 Random Forest

While there has been an expansion of deep learning models (and we attempt as well) within natural language processing, mathematics student open responses are not necessarily comparable to the corpus in most prior analyses. For example, the datasets made available through competitions (cf.[15]) have largely focused on non-mathematics content to which others have been able to explore a range of methods including that of deep learning[24]. However, the differences in data sources may be worth noting in comparing this to prior works. Namely, many deep learning methods, particularly those using pre-trained embedding models as we describe later in this work, are unable to effectively represent numbers and equations well in the context of other words; while such representations recognize some numbers, the corpus is limited. It is for this reason that a bag-of-words type of approach begins to make more sense as

Table 1: Sample Responses from Example Problem Selected from Illustrative Math open educational resource

Grade	Example Responses
5	Because B is 2x biggest than A
4	I didn’t understand ?
5	Because 2/12 time 2 equals 5
5	2.5 x 2=5
5	2.5 times 2 is 5 so the scale factor is 2 oops that is what I meant
5	Cause 2.5 divided by 5 is 0.5
3	Because the top one is 2.5 and 1.5 goes to 2.5
1	I guessed
3	Because the part on a is half the size of the one on part b.
5	2.5 times 2 is 5.
5	A has 2.5 on top and B has 5_2.5 x2is withc means that it was 2
2	I said that because two of them are equal

Table 2: Example Tokenization: Standard Count Vectorizer

Raw Student Text	Count Vectorizer Tokenizater	Stanford Tokenizer
"The answer couldn't be 6x4 = 6" "skies are blue" "It's something I dont understand" "I didn't know, so is this right? x+4=8 6x1"	["6x1", "6x4", "answer" "are" "be", "blue", "couldn", "didn" "dont", "is", "it", "know" "right", "skies", "so", "something", "the" "this", "understand"]	["The", "answer", "could", "n't" "be", "6x4", "=", "6" "skies", "are", "blue", "It" "'s", "something", "I", "dont" "understand", "I", "did", "n't" "know", ",", "so", "is" "this", "right", "?", "x" "+4", "=", "8", "6x1"]

it is easier to train such a method to recognize all words within our specific context. Additionally, the tree-based methods likely require fewer training examples than a complex deep learning model but still offer a large degree of non-linearity in their representations of data.

In regard to the random forest model explored in this work, as discussed earlier, the input of this model is the term frequency inverse document frequency value. This assists in lowering the weight of less important stop words. We allowed the forest to contain 100 decision tree's. By having a more slightly more robust dataset, there is less of a chance of over fitting. Additionally, this allows the forest to identify as many important words within the student responses. For each of the 100 trees, pruning is not performed. This allows for each of the trees to expand out and identify as many words as impactful. Once again, this does bring in the risk of overfitting. Training and testing is performed with a 10-fold cross validation. The model then output a probability that the grade would belong to each of the 5 categories. These probabilities are then used as covariate within the final Rasch model described later in this section.

5.2 XGBoost

Continuing with the tree approachs, XGBoost was another flexible model applied. This method will apply gradient boosted decision trees. As [2] describe, there are three parts to the tree boosting. First, a regularized learning objective is calculated to prevent overfitting. It starts by calculating a prediction thru summing all the independent tree structures and leaf weights from ensembled decision trees. From there, the model attempts to understand what were the effective set of functions learned within the model by minimizing the loss function. This function is calculating the difference between the predictions and the targets, while attaching a complexity penalty. By attaching this penalty, as the authors discuss, the final weights of the ensembled trees are smoothed to avoid overfitting.

From there, the model aims to optimize the ensembled trees, but the authors [2] noted that with functions as the parameters, a traditional euclidean space is not able to be used for the optimization. This lends itself to an additive modeling approach by adding more functions and calculating the loss function. The model adds the functions which minimise the loss function and most improves the current model.

Additionally, the model aims to combat overfitting by utilizing shrinkage, also commonly referred to as regression to the mean. As the authors note[2], by utilizing the shrinkage, it can help to reduce how much influence each tree has on the overall ensembling. This then can help create more room for additional, potentially stronger trees, thus improving the model while reducing the chances of overfitting. Lastly, the XGBoost takes one aspect from the Random Forest model, and that is feature subsampling.

Similar to the Random Forest, the term frequency inverse document frequency matrix is used as input to the model. We set the

model to perform multi-class classification with a softmax probability learning task. This then allows the model to produce a separate probability for each possible grade. Once again, all training and testing is performed using 10-fold cross validation.

5.3 LSTM

The final student grade prediction model for comparison is a deep learning long-short term memory (LSTM) network [7]. As mentioned previously, deep learning has been on the forefront of recent advancements in natural language processing. Such models differ from the more traditional methods described above in that such networks consider the ordering of words. Contrary to approaches using a bag-of-words approach, LSTMs recognize that the ordering of words may contribute to the interpretation of the responses and considers this within the network structure.

Before modeling, each sentence is first processed to remove unwanted characters such as line endings. Next, stop words are removed from each sentence to help reduce the sentence to only the most representative words; by shortening the sequence of words it is also believed that the model will be able to more efficiently learn from the data in that it will not need to learn to ignore such common words. From here, each sentence is tokenized using the Stanford Tokenizer and subsequently vectorized using the pre-trained 100-dimensional GloVe embeddings described in Section 4.1.

We apply a bi-directional LSTM model consisting of 3 layers: a 100-dimensional input layer that accepts the pre-trained GloVe vectors of each word, a 40-node hidden LSTM layer (20 nodes that observe the sequence in order and 20 nodes that observe the sequence reversed), and finally a 5 node output softmax layer corresponding to each of the 5 possible grade values with a cross-entropy loss applied. The application of a bi-directional network is believed to help the model learn order dependencies between words as well as help it learn more prominent long-term dependencies at the beginning and end of each response.

The model is trained using a 10-fold cross validation with an Adam optimizer with a step size of 0.03. The small step size combined with the comparably small network size (it is not uncommon for such networks to contain many more layers with hundreds of nodes per layer) is meant to help reduce model overfitting; while the overall dataset is arugably large enough to support deep learning models, a separate model is trained per problem which, in some cases, exhibit smaller sample sizes than would normally support a deep learning model. However, given the model size and the pre-trained nature of the GloVe embeddings (the model does not need to learn new word representations), we feel that the application of this model is justified for the given prediction task.

The model is trained using a variable stopping criterion based on the performance of a holdout validation set consisting of 1-fold's worth of data (approximately 1/9th of the available training data). The model is trained over many epochs, or cycles through the training data, until the performance on the validation set plateaus.

Similar to the other previously described models, the LSTM treats each grade as mutually exclusive classes for training; despite the ordinal nature, this aspect of the output is not explicitly included in the model

5.4 Rasch Model

While the previously described machine learning and deep learning models are the focus of comparison for this work, we utilize one final model as a baseline and a means of more fairly evaluating the performance of the previous models. For this, we use a two- and three-component Rasch model. A Rasch model, commonly applied in item response theory (IRT), is a probabilistic model (in this case, a variational bayes model) that uses fully connected data to learn components that describe the users and content independently of each other. In IRT, it is common that such a model may learn a student ability parameter for each student as well as an item difficulty parameter describing each problem. In our specific application, we use the Rasch model to learn a student ability parameter, item difficulty parameter, and, for an additional comparision, a teacher strictness parameter following the results of our pilot study described in Section 3.

The formulation of the Rasch model is as follows:

$$grade = ordered\_logistic(student\_ability - item\_difficulty + \beta * X) \tag{1}$$

$$grade = ordered\_logistic(student\_ability - item\_difficulty - teacher\_strictness + \beta * X) \tag{2}$$

In the first Rasch model in Equation 1, it is formulated as an ordinal logistic regression observing a learned value per student and a learned value per problem. In addition to this, a set of covariates X will be used to evaluate the previously described machine learning models. Since the base Rasch model, where X is an empty matrix, observes no information of the problem text itself, a model that is able to effectively learn from student response text should lead to notable improvements in model performance. It is for this reason that we use this model as a means of comparison. The predictions of each of the previous models will be incorporated into the Rasch systematically to compare the added benefit (if any) beyond the attributes of student ability and item difficulty. For example, the 5 predicted probabilities produced by the LSTM model are presented to the Rasch model as X and the performance of such a model is compared to the Rasch model without such covariate data (as well as compared to the Rasch model containing the other machine learning predictions).

The Rasch model in Equation 2 incorporates an additional learned parameter of teacher grading strictness, in observance of the results of our pilot study. By including this term, we should gain an understanding of how well such a model is able to perform when observing that different teachers grade with different policies, particularly in regard to being more or less strict (i.e. a less-strict grader may apply higher grades on average than a more-strict grader).

In both of these cases, the Rasch model helps to observe the model performance independent of student "goodness" and item difficulty that may otherwise inflate or deflate model performance. In addition to this, the Rasch incorporates an ordinal regression which was not observed by any of the other machine learning models; as such, the combination of the two methods holds promise

Table 3: Rasch Model Performance

Model	AUC	RMSE	Kappa
Rasch Model with teacher component	0.696	1.09	0.162
Rasch Model without covariates	0.827	0.709	0.370
Rasch Model with number words covariates	0.829	0.696	0.382
Rasch Model number words and Random Forest covariates	**0.850**	**0.615**	**0.430**
Rasch Model number words and XGBoost covariates	0.832	0.679	0.390
Rasch Model number words and LSTM covariates	0.841	0.637	0.415

to produce better results by observing the ordered relationship between grades.

As one additional baseline model, we include the Rasch model with a single covariate representing the number of words in the student response. It seems plausible that longer responses may, on average, receive higher grades, so we include this term alongside the others as a more appropriate baseline of comparison.

6 RESULTS

We report three evaluation metrics with which to compare each model: AUC, RMSE and Cohen's Kappa. AUC is calculated using a simplified multi-class calculation of ROC AUC [5], where values close to 0.5 represent performance at chance and values close to 1 represent higher performance. RMSE is calculated as the root of average squared errors when observing the ordinal predictions and labels (i.e. observing that the difference between a prediction of 3 and an actual grade of 4 is a value of 1); this differs from the other metrics that observe the 5-point labeling scale as a multi-class classification problem. Finally, we observe multi-class Cohen's kappa as a measure of inter-rater agreement above random chance (observing that some labels such as 4 and 0 appear more frequently than others.

Overall, each of models managed to predict student open response better than the simple Rasch model baseline. The baseline model, of just a basic Rasch model without any additional covariates, managed to classify students' open response grades with an AUC of 0.827, as shown in Table 3. In fact, all models manage to classify student grades with an AUC greater than 0.820 aside from the Rasch model incorporating a teacher strictness component; it is possible that the model is either unable to learn three parameters from the given data or the influence of variations in teacher grading are not as impactful as the pilot study suggested. Likewise, aside from the identified Rasch model, the Kappa values are moderately high, all models are able to classify and predict the student's grade at least 37% above chance.

However, its is apparent that the incorporation of the machine and deep learned grade prediction covariates provide the Rasch model with insight previously not identified. While the LSTM and XGBoost manage to improve the models performance, Random Forest managed to provide the most additional insight to the Rasch model. What is also evident is our model's ability to become more confident in our predictions with more covariates. Our RMSE manages to drop in the Rasch model with any of our additional covariates and, once again, the Random Forest managed the lowest RMSE.

In the end, it is clear in Table 3 that the best overall model was the Rasch model with Random Forest covariates. This model was able to classify/predict student's open response grades with an increase in the AUC of 0.023, a drop in error rate (RMSE) of 0.094, and an increase in the kappa of 0.060 over the baseline Rasch model without additional covariates.

7 EXPLORATION OF SAMPLE SIZES

While the results in the previous analysis suggest that the models are performing moderately well in comparison to our baseline, this research aimed to explore the impact the amount of data has on our performance. It is unclear if, given more data, we would expect to see large increases in model performance. We selected a problem from our dataset to exemplify this process here, but it is intended that this analyses can be repeated on all problems to assess the impact of available data at a finer level of granularity. Of the 10 problems with the closest grade distribution to that of the overall population, we selected the problem with the largest sample size (shown in Figure 2).

This last analysis was performed with a leave one out cross validation and an increasing training set sample size. Starting at 5 training points, we train and predict the test point. We repeat this sampling 10 times to allow us to calculate our confidence intervals. Following the 10th iteration, the sample size is increased to 15, and the process repeats for the same test point. This is repeated, increasing the sample size by 10 until it can't sample anymore. Once this is finished, the model moves to the next test point of the leave one out cross validation and repeats. In this way, we create a bootstrapping example of how model performance changes at each sample size; where we see the model performance stabilizing and beginning to plateau, it is suggestive that additional data would not lead to substantial gains in model performance. For this analysis, we used the random forest model alone without the Rasch for exemplary purposes.

In terms of the sample size and its effect on our ability to predict a student's open response grade, it is clear from Figure 3 and the confidence that we see a statistically significant improvement in the performance from sampling 5 training points to just under 55 training points. However, what is evident is that the model has maxed out its potential in its current form at just under 55 training points, and that additional data is not significantly improving the ability to predict the student's grade. With plots such as Figure 3 it suggests that any further improvement's would require updates to the model and data representation rather than simply collecting more data.

Figure 3: Selected for exemplary purposes, this plot illustrates the random forest model performance within a single problem over increasing sample sizes. Similar plots are generated for every problem to observe where our models may benefit from more data.

8 DISCUSSION

Overall, the study aimed at utilizing modern machine and deep learning approaches to predict grades from authentic student open responses. With the ensembling of machine and deep learning with Rasch models, we have shown a strong ability to predict a students grade. Additionally, this study showed that in some cases more data wouldn't necessarily improve performance. Thus providing us the understanding that our ability to predict a grade, for a specific problem, may or may not improve with more data. However, given there are 2,042 unique problems, a limitation of this part of the study is that it's difficult to ascertain this information for each individual problem.

9 FUTURE WORK

With the overall strong model performance, there are a couple next steps we wish to address in future work. There is still a weakness in our model's ability to understand text. Currently, the best performing model, the Random Forest, is utilizing a bag of words approach, counting the words within student responses. There is no consideration of the structure of the student's response or what words relate to other words within the student's response. We attempt to combat this hindrance by utilizing the LSTM, but the lesser results of that model in Table 3 suggest that either the semantics and context of the words did not provide additional insight to the model, or, more likely, there is not enough data within each problem for such a model to effectively learn. The representation of data may also be an issue across these models, specifically in reference to the pre-trained GloVe embeddings. While a very powerful tool, as shown in previous research and discussed in this paper, models which utilize this are bound to the words in the pre-trained corpus. Even with the use of a Wikipedia trained GloVe embedding, our LSTM did not gain much in terms of additional information. Understandably, many functions and formulas students write aren't represented in the pre-trained embeddings. Currently, our team is developing an approach to expand these pre-trained embeddings to account for missing words, functions or math terms without requiring re-training of a GloVe embedding.

It is our goal to use these findings and the continued development of these grading models to deploy tools that can help teachers save time in assessing their student work so that they may direct their attention to the students who would most benefit from additional feedback.

ACKNOWLEDGMENTS

We thank multiple NSF grants (e.g., 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, 1535428, 1440753, 1316736, 1252297, 1109483, & DRL-1031398), the US Department of Education Institute for Education Sciences (e.g., IES R305A170137, R305A170243, R305A180401, R305A120125, R305A180401, & R305C100024) and the Graduate Assistance in Areas of National Need program (e.g., P200A180088 & P200A150306), and EIR the Office of Naval Research (N00014-18-1-2768 and other from ONR) and finally Schmidt Futures.

REFERENCES

[1] Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education* 25, 1 (2015), 60–117.
[2] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 785–794.
[3] Wael H Gomaa and Aly A Fahmy. 2012. Short answer grading using string similarity and corpus-based similarity. *International Journal of Advanced Computer Science and Applications (IJACSA)* 3, 11 (2012).
[4] Arthur C Graesser, Peter Wiemer-Hastings, Katja Wiemer-Hastings, Derek Harter, Tutoring Research Group Tutoring Research Group, and Natalie Person. 2000. Using latent semantic analysis to evaluate the contributions of students in AutoTutor. *Interactive learning environments* 8, 2 (2000), 129–147.
[5] David J Hand and Robert J Till. 2001. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine learning* 45, 2 (2001), 171–186.
[6] Neil T Heffernan and Cristina Lindquist Heffernan. 2014. 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 (2014), 470–497.
[7] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
[8] Thorsten Joachims. 1996. *A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization.* Technical Report. Carnegie-mellon univ pittsburgh pa dept of computer science.
[9] Kelly YL Ku. 2009. Assessing students' critical thinking performance: Urging for measurements using multi-response format. *Thinking skills and creativity* 4, 1

(2009), 70–76.

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

[11] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. 55–60.

[12] Michael E Martinez. 1999. Cognition and the question of test item format. *Educational Psychologist* 34, 4 (1999), 207–218.

[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[14] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[15] Automated Student Assessment Prize. [n. d.]. The Hewlett Foundation: Automated Essay Scoring.

[16] Brian Riordan, Andrea Horbach, Aoife Cahill, Torsten Zesch, and Chong Min Lee. 2017. Investigating neural architectures for short answer scoring. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. 159–168.

[17] Carly Robinson, Michael Yeomans, Justin Reich, Chris Hulleman, and Hunter Gehlbach. 2016. Forecasting student achievement in MOOCs with natural language processing. In *Proceedings of the sixth international conference on learning analytics & knowledge*. ACM, 383–387.

[18] Jeremy Roschelle, Mingyu Feng, Robert F Murphy, and Craig A Mason. 2016. Online mathematics homework increases student achievement. *AERA Open* 2, 4 (2016), 2332858416673968.

[19] Mark G Simkin and William L Kuechler. 2005. Multiple-choice tests and student understanding: What is the connection? *Decision Sciences Journal of Innovative Education* 3, 1 (2005), 73–98.

[20] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. *arXiv preprint arXiv:1506.06714* (2015).

[21] Jana Zuheir Sukkarieh and John Blackmore. 2009. c-rater: Automatic content scoring for short constructed responses. In *Twenty-Second International FLAIRS Conference*.

[22] Jana Z Sukkarieh, Stephen G Pulman, and Nicholas Raikes. 2003. Automarking: using computational linguistics to score short, free text responses. (2003).

[23] Md Arafat Sultan, Cristobal Salazar, and Tamara Sumner. 2016. Fast and easy short answer grading with high accuracy. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1070–1075.

[24] Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 1882–1891.

[25] Siyuan Zhao, Yaqiong Zhang, Xiaolu Xiong, Anthony Botelho, and Neil Heffernan. 2017. A memory-augmented neural model for automated grading. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*. ACM, 189–192.