Is It Fair? Automated Open Response Grading

Anonymous
Anonymous Institution
anonymous@anonymous.edu

ABSTRACT

Online education technologies, such as intelligent tutoring systems, have garnered popularity for their automation. Whether it be automated support systems for teachers (grading, feedback, summary statistics, etc.) or support systems for students (hints, common wrong answer messages, scaffolding), these systems have built a well rounded support system for both students and teachers alike. The automation of these online educational technologies, such as intelligent tutoring systems, have often been limited to questions with well structured answers such as multiple choice or fill in the blank. Recently, these systems have begun adopting support for a more diverse set of question types. More specifically, open response questions. A common tool for developing automated open response tools, such as automated grading or automated feedback, are pre-trained word embeddings. Recent studies have shown that there is an underlying bias within the text these were trained on. This research aims to identify what level of unfairness may lie within machine learned algorithms which utilize pre-trained word embeddings. We attempt to identify if our ability to predict scores for open response questions vary for different groups of student answers. For instance, whether a student who uses fractions as opposed to decimals. By performing a simulated study, we are able to identify the potential unfairness within our machine learned models with pre-trained word embeddings.

Keywords

ACM proceedings, LaTeX, text tagging

1. INTRODUCTION

In recent years, natural language processing (NLP) has been at the forefront of machine learning in multiple fields. Whether it be within corporations or within the scientific community, NLP has provided deeper insights into consumer and user behaviors. Linguistics provides another source of information outside the standard data from user logs. Instead of relying on correlational assumptions from this data, inferences can be deduced directly from the users linguistics. While utilizing linguistics in education isn't genuine, modern machine learning and natural language processing has helped to automate the analysis and provides effective tools for learning. Especially within the online educational technology environment.

While online educational technologies has embraced linguistics, more specifically linguistics of teachers, students and chat systems; in recent years, the development of more advanced deep learning has brought a deeper semantic understanding of words within these linguistical models. More specifically, there has been a rise in sequential models which utilize word embeddings and vector spaces to develop algorithms which understand the semantic meaning of the words in sentences, to be able to infer more accurate predictions.

With the emergence of word embeddings and their vector spaces, many researchers have looked to utilize these approaches in their analysis in multiple fields. However, there is one shortcoming of word emnbeddings; to develop an accurate word embeddings which allows for accurate semantic understanding of words (based on their location within the embedding vector space), a researcher requires copious amounts of data. Without these large datasets, the vector spaces may provide very inaccurate semantic relationships of words. Thus, companies and universities sought out to utilize their own, or crawl the internet for their own, larger datasets to generate their own word embeddings. They would then publish them for public usage.

The emergence of word embeddings was an important development in machine learning and NLP, but the publishing of publicly available pre-trained word embeddings provided researchers with a powerful tool for optimizing algorithms with linguistics. While word embeddings were powerful for studies within areas such as MOOCS (i.e [13][19]), smaller studies, with less robust linguistic data, were unable to utilize this modern approach for semantic relationship of words. Pre-trained word embeddings cut through that by providing researchers with more robustly trained word embeddings. Thus, researchers had a vector space which allowed for semantic relationships of words which their algorithm wouldn't have been able to generate on their own.

Its undeniable that having a word embedding trained on larger datasets, such as GloVe[20] being trained on data from

Wikipedia or Word2Vec[18] being trained on all of GoogleNews, provides deeper insights for the algorithm into the semantic meaning of words; research has shown that the language which those embeddings were trained on provided underlying known biases [2]. For instance, Word2Vec, as mentioned earlier, was trained on GoogleNews. The language utilized influenced the word embeddings to relate words in a bias way.

Since research has shown that some of the semantic meanings inferred from pre-trained word embeddings can elicit undesirable biases, the major question then becomes, does this underlying bias suggest the algorithm or predictive model will make unfair decisions? For instance, if an algorithm utilizes linguistics and NLP with pre-trained word embeddings will the predictions be unfairly made from those underlying biases (i.e. a scoring mechanism changing scores for certain groups of students). Our research attempts to explore:

1. Whether, through 3 simulated studies, the format a student writes an answer (i.e. fractions vs. decimals) effect the scoring model and potentially elicit unfair scoring?
2. What effect, through 3 simulated studies, if any, do *'distractor'* words have on the unfairness?
3. Whether or not underlying bias in pre-trained word embeddings can lead to unfairness in open response scoring models in middle school mathematics?

The simulated study and the analysis of the genuine middle school mathematics data utilize the recently published approach, termed ABROCA [7], to evaluate what level of potential unfairness is present.

2. BACKGROUND

2.1 Online Educational Technologies: Intelligent Tutoring Systems

In recent years, online educational technologies have been on the forefront of learning for students. While most learning with these systems have been supplemental to in-person learning, most recently, these systems have become more relied upon to deliver an effective education. A common online educational technology, intelligent tutoring systems (ITS) [4], has been prevalent in education for many years. Some of the most common ITS are ASSISTments[10], McGraw Hill's ALEKS^{TM} and/or Carnegie Learning's Cognitive Tutor^{TM}. These systems aim to support both students and teachers through automated summaries of student performance, automated feedback and grading to students, hints, scaffolding, and common wrong answer messages. Through the use of both machine learning and software engineering, these systems have been shown to be effective at increasing the scores of students with end of the year standardized math exams[24] and the effects of their intelligent tutoring closely resembles the effect face to face tutoring has on students[30]. Other ITS, such as AutoTutor[8],have attempted to resemble the face to face tutoring more directly by developing automated conversations and dialogues between students and ITS [8]. However, most of the support and benefits of these ITS have been limited to questions with structured answers (i.e. multiple choice or fill in the blank questions).

2.2 Automation of Intelligent Tutoring Systems

While there are a plethora of ITS offering automated support for both students and teachers, this is mostly limited to questions with structured answers. Mainly, multiple choice or fill in the blank. It should be noted, that some of these systems, such as ASSISTments, support open response or short answer questions. However, the automation is limited to questions with structured answers. For a system such as ASSISTments, McGraw Hill's ALEKS^{TM} and/or Carnegie Learning's Cognitive Tutor^{TM}, it is straight forward to teach a system that A is the correct answers. Thus, if a student selects B, a system can easily grade and suggest formulated feedback to that selection. The answers are finite.

Automated support of ITS is a draw for many teachers; one study noted that many utilize multiple choice questions for the efficiency and accuracy of grading [25]. However, since most of the automation is limited to questions with structured answers, the content which teachers provide is limited. To be able to expand the system's automation purview, natural language processing (NLP) has been brought to the forefront. Studies have looked to utilize NLP to automatically evaluate work or questions which require a student's unique linguistics (i.e. open response questions, or essays) including [28][27][23][1][6][29][16]. While most of this research has been primarily focused on content outside of mathematics, our previous research, [Blinded for Review], looked to help teachers diversify the content which they provide students in middle school mathematics by utilizing traditional and modern NLP to develop an automated scoring model for open response middle school mathematics questions. A more diverse set of question types can be beneficial to students and can elicit differing levels of cognition, as studies [17][14] note.

2.3 Natural Language Processing

Towards the goal of automating open response questions, or any linguistical/NLP prediction task, the major task is in how to numerically represent words thus that a machine learned algorithm can generate an accurate prediction. One of the more simplistic approaches utilizes the frequencies of each unique word within the corpus, whats commonly known as a *Bag of Words* approach. While undoubtedly easy to interpret and not computationally intensive, this approach has been utilized in studies such as [12] and is the foundation of more advanced approaches such as[26][9].

While frequency based approaches, like bag of words, are simplistic in nature and can provide insight, a major pitfall is that they begin to weight words more that occur more frequently. However, words occurring most often aren't always the most important or most informative. A common approach to combating this is to utilize term frequency inverse document frequency (TF-IDF). One study was able to use TF-IDF to accurately match words written in a query to the documents that are the most closely related[21].

Eventually, with the advancement of machine learning and deep learning, more modern approaches have gone to utilizing embedding vectors to represent words. Essentially, each word will have an attributed list of numbers which places that word in the embedding vector space. From this vector space, deep learning can utilize their locations within the vector space to understand the semantic relationship of words. As mentioned earlier, GloVe[20] and Word2Vec[18] are two of the most common word embedding algorithms. However, for these approaches to be effective, there needs to be enough data present to generate the proper semantics of words. Without enough data, it is likely any semantics are ill identified and the embedding space is ill defined. From this, it is difficult for an algorithm to utilize the generated embedding space to accurately understand what text means or what it is inferring.

2.4 Pre-Trained Models

While word embeddings have been some of the most prevalent NLP techniques in recent years, there is a hindrance to this approach, data. To develop an accurate word embedding, with accurate informative semantics of words, there needs to be enough data with robust enough text. As mentioned earlier, if there isn't enough data, incorrect semantics of words can be inferred and the algorithm will incorrectly interpret text and linguistics. However, efforts have been made to combat this through pre-trained models. Instead of generating word embeddings from scratch, those with access to larger corpuses and datasets, such as Google and Stanford, trained their own Word2Vec and GloVe embeddings on GoogleNews and Wikipedia, respectively. This undoubtedly provides researchers with a very powerful asset to their NLP. Now, researchers can use pre-trained word embeddings with smaller corpuses and develop predictive models with embeddings generated from datasets that dwarf their own. This means a study which wouldn't be able to accurately utilize word embeddings in their predictive model, now can. As these pre-trained word embeddings have grown in popularity, word embeddings have expanded to utilize bidirectional encoder representations from transformers (referred to as BERT[5]) to create pre-trained word embeddings, as well.

With the success of word level embeddings, researchers looked to develop sentence level embeddings. Similar to the word embeddings, sentence level embeddings utilize deep learning to generate embedding vector spaces and embedding vectors which represent entire sentences. Two common approaches used are SBERT[22] and the Universal Sentence Encoder[3] (often referred to as USE). Additionally, embeddings have expanded from the word and sentence level to document level embeddings. Approaches, such as Doc2Vec [15], are able to generate a single vector representation of entire documents. These more generalized embeddings allow for simpler direct comparisons of sentence and documents versus individual word embeddings. Similar to the word embeddings, these approaches are often pre-trained and released for public use.

2.5 Fairness

There are clear advantages to word embeddings and even more advantages to pre-trained word embeddings. This is also clear with sentence and document level embeddings as well. As discussed earlier, not everyone will have the resources to pull and analyze massive datasets to be able to accurately generate embeddings at the word, sentence or document level. With Google utilizing GoogleNews and Stanford utilizing Wikipedia, researchers have the opportunity to utilize semantics where they wouldn't have been able to previously. However, all of these pre-trained algorithms begs the question, what is being inferred from the data which is was trained on?

When it comes to linguistics, the way someone speaks, the way someone articulates can be unique to themselves. Similarly, the way someone writes is personal to themselves and specific to their topic. So when algorithms are being pre-trained on data which isn't the researchers own data, there are questions to be asked. For instance, while there is more data, what are some of the semantic relationships these embeddings are identifying? From the word level, if the embeddings are developed from GoogleNews or Wikipedia, what is being identified? A recent study [2] looked to identify the semantic similarities.

Research[2], has been able to identify some potentially harmful semantic relationships present in common pre-trained word embeddings. For instance, [2] was able to identify that Google's pre-trained Word2Vec on GoogleNews elicited some harmful stereotypes. As the title of their research states, Google's pre-trained Word2Vec on GoogleNews closely associates ***Man*** with ***Computer Programmer*** and ***Woman*** with ***Homemaker***. Similarly this study looked to see what other potential gender stereotypes could be present within these pre-trained word embeddings. The authors managed to see that, for instance, occupations most closely related to the pronoun ***She*** were nurse, receptionist, socialite, housekeeper, nanny; and the occupations most closely related to the pronoun ***He*** were maestro, captain, skipper, boss and protege, just to name a few. There is clear evidence, that the language used within GoogleNews perpetuates certain stereotypes and undesirable biases.

While its clear that undesirable bias and harmful stereotypes are present in the pre-trained word embeddings, it doesn't guarantee that predictive models which utilize these are inherently biased. It may be the case that the algorithm could potentially be inferring dangerous semantic relationships, but is it effecting the decision the algorithm makes. In education, this is needs to be explored deeper. Automated scoring algorithms should be developed with the intention of scoring students without bias or harmful stereotypes being considered. That's why, for instance, in our past research [Blinded for Review], all demographics were left out of the automated scoring model. It was our goal to score the student on their content; and content alone. There in lies the important question, while omitting variables which could cause unfairness in the automated scoring, are we continuing to avoid unfairness if we utilize pre-trained word embeddings.

Naturally, the next question becomes, how does one identify potential unfairness in their algorithms or predictive models? For instance, how can you identify if an open response answer automated scoring model is unfairly scoring? Recent work,[7], developed an approach called Absolute Between-ROC Area (ABROCA). This approach utilizes the areas between two ROC curves to identify a model's ability to perform a classification task between two different groups of data. For instance, with open response answers, some students may write answers with mostly fractions and another group of students may use decimals and fractions. By generating the ROC curve of the prediction task for each group, you can utilize the area under the curves to identify the potential unfairness. So if there is a small area between the two ROC curves, one for the prediction task for each group, the less unfairness. However, if there is a large area, there is more unfairness in the prediction model. This research aims to develop a simulated study to examine whether the utilization of the pre-trained GloVe word embeddings within an automated open response scoring model can elicit unfair scoring, and whether or not there is evidence of unfairness with our previously developed open response scoring model in middle school mathematics by utilizing ABROCA as the evaluation metric.

3. STUDY 1: SIMULATED STUDY OF FAIRNESS IN AUTOMATED SCORING

It is clear that embeddings have become an integral part of NLP and those hoping to develop predictive models utilizing linguistics are often drawn to the semantic properties which your model can utilize and learn. While researchers have noted that pre-trained embeddings are very powerful in providing an embedding vector space developed from robust datasets, and its clear there are undesirable biases built into those embedding vectors, its unclear as to whether or not those undesirable biases or stereotypes influence algorithms unfairly. Thus, this research developed a simulated study to attempt to identify if pre-trained word embeddings are utilized within an automated scoring model for open response answers, do they influence the model to make unfair predictions. As mentioned earlier, an example of this would be if a group of students states their answer with a fraction and surrounding text, does the predictive model generate scores similarly for those students that use decimals along with surrounding text? Through this simulated study, we are able to gain a deeper insight into what/if any unfair scoring occurs when utilizing the pre-trained GloVe word embeddings trained on Wikipedia.

There are 3 studies within this simulated study to help achieve this goal. First, we develop answers which contain differing distributions of answers which contain fractions and decimals and generate the ABROCA value at the differing distributions. Second, we attempt to see if decimals and fractions alone generate differing ABROCA values. Third, we attempt to see if additional 'distractor' words replace decimals in the text, do the ABROCA values differ at differing distributions? These studies will help provide deeper insight into the potential unfairness an automated scoring model can be producing when utilizing pre-trained word embeddings

3.1 Data Generation

At the foundation of this simulated study is the generation of the *student* dataset. The goal of this process was not to just generate 4 or 5 unique answers and randomly select 100 of those. This study set out to generate answers with more variability in their content and linguistics. To accomplish this, the generation was split into to facets, the training dataset student answers and the test set student answers. This was performed such that the model would not be able to have any identical answers between the training set and the test set. While this does create more noise, it helps to isolate what correlations our scoring model will eventually identify and predict from. Essentially, that the predictions aren't being made because the model has already seen that exact series of embeddings associated with a certain score.

3.1.1 Training Data: Corpus Generation

Table 1: Sample of Phrases and Their Associated Avg. Score

Generated Phrases	Avg. Score
my answer is	0.718750
i picked	0.622222
i guess the answer is	0.600000
i think it is	0.600000
i think the answer is	0.590909
i worked out	0.585366

Towards the goal of generating student answers with enough variability in their content, the generation of the corpus was founded on the goal of utilizing random selection. From this randomization, this study can mimic real open response student answers. This was based on the intuition of what makes open response answers unique is the variability within the answers. In our previous study [Blinded for Review], we were able to infer that many answers were similar, but not fully identical. First, as Table 2 shows, there are 4 different length student answers in this corpus. There are answers which are 6, 5, 4 and 3 word length answers. The generation of the student answers can be surmised into 4 steps and visualized with Table 2:

1. Select whether it will be a student answer which uses decimals or fractions
2. Randomly select what length the answer is.
3. Once a length is randomly selected, another random selection is made between the two structures (i.e. 'Answer Structure' in Table 2)
4. Randomly select text from **Fill "1"** and **Fill "2" Fractions** or **Fill "2" Decimal** to fill the identifiers **'1'** and **'2'**

To summarize, the first step of the generation of the student answers is to decide whether the answer will contain a decimal or a fraction. This is followed by randomly selecting what length the answer is. Once a selection is made, there are two potential answer structures to choose from. In Table 2, this is the column 'Answer Structure'. For all length answers, there are two types of answers with different structures. Another random selection is made between the two structures. From there, *'1'* and *'2'* are filled with random selections made from the available text (i.e. *Fill "1"* and *Fill "2" Fractions* or *Fill "2" Decimal* in Table 2). Another way to describe the text and language used in *Fill "1"* are 'distractor' words.

3.1.2 Test Data: Corpus Generation

With a corpus generated to simulate training data, the next step includes generating a testing corpus of student answers to select from. The steps are the same as the generation of training dataset steps listed above. However, Table 3 shows there is one key distinction between the training and test generation, the text which can be filled (*Fill "1"*). More specifically, the answers which are generated for the test set will never occur in the training set. As mentioned earlier, this was performed for two reasons. One, this allowed for a more realistic distribution of student open response answers. Often times, answers are similar, but few are identical. This is what makes automatically scoring open responses questions difficult. There are a infinite set of answers. Therefore, this variability helps to simulate data genuine student answers. Secondly, by having different text and phrases to select from that are different than the training set corpus, guarantees that our automated scoring model will not be identifying sequences of words, or phrases, that are identical in the training and test set. This allows us to understand, more specifically, what our algorithm is making decision on and what correlations its finding. If it see's the exact same answer it was trained on and predicts a score, that doesn't provide insight to whether the word embeddings are impacting the fairness of the algorithm, rather that it has identified an identical answer. Without this step, it would be difficult to identify if any changes in predictability between two groups are from one group having the identical answers and scores, the 'distractor' words, or the math terms.

It should also be noted that the *Answer Structure* for the test corpus is also different than the training corpus. In the training dataset, words were being selected and placed in the answers for the 'distractor' words. Whereas in the test dataset, whole phrases are being selected for the 'distractor' words. Again, this allows for variability in the answers between the training and test datasets.

In the end, a separate corpus of student answers was generated which contain decimals and fractions, separately. Therefore, an individual corpus of generated student answers using fraction for both training and test and an individual corpus of generated student answers using decimals for training and test datasets were generated. These corpuses are what will be used to select the final training and test data.

As for the scoring of the simulated student answers, a general rule was set that any answer that contains 3/4 or 0.75 is considered correct. All other answer are considered wrong. Partial credit is not considered in this simulation study. Thus this is a binary classification task.

3.2 Methodology

Once the corpuses have been generated, the process of selecting data can begin. This can be surmised by the overall goals of this study. This study sets out to identify if a automated scoring model for open response questions, which utilizes pre-trained word embeddings, elicit unfair scoring. To accomplish this analysis, there needs to be an identifiable difference, outside of the 'distractor' words, between student answers. In this case, each student open response answer has 'distractor' words and either a decimal or a fraction (as discussed in the previous section). Inversely, our goal of this simulation study is to also extrapolate whether the 'distractor' words, not the fractions or decimals, influences any unfair tendencies in the scoring of the simulated student open response answers.

For the sampling of the simulated student open response answers, we set out to simulate data which consists of a balance of student answers which utilize fractions and decimals. The training set is made up of simulated student answers from both the answers which contain fractions and contain decimals separately. The steps to the selection process is as follows, at an instance a simulated student answer is to be selected:

1. a student answer is always drawn from the training dataset of simulated answers which contain a fraction. This is considered *Group A* students.
2. a random integer is drawn
3. if the integer is below our specified threshold, another selection is made from the training dataset of simulated answers which contain a decimal. This is then considered an answer from *Group B* students.
4. if the integer is above our specified threshold, another selection is made from the training dataset of simulated answers which contain a fraction. This is also considered an answer from *Group B* students.

A threshold was set for selecting decimals and fractions to control the balance of answers. This lends itself to our goal of being able to identify whether or not the format a student writes an answer, i.e. using factions vs. decimals, effects our ability to score student open response answers. By having a threshold, we can increase the threshold incrementally and see what is the model's ability to score the simulated student answers. So as the threshold increases, more and more answers that contain decimals (Group B students have more and more answers containing decimals) are selected and trained upon. Thus, with ABROCA, fairness can be identified.

For the test set, a similar approach is taken. Since the training set contains answers of both Group A students, which are students who all answered with a fraction in their text, and Group B students, which some student used fractions and some used decimals, the test set will contain the same Group A distribution of answers containing fractions only, but with different content making up those answers, and the same Group B distribution of answers containing decimals.

While it was emphasized that the training and test sets have similar distributions of answers containing decimals and fractions, the two datasets have identical distributions of grades. This was done to remove outside influence on the automated scoring model. If there is an unbalanced grade distribution, then the performance of the model could be driven by more scores of 0.0 or 1.0. By balancing the grades across both the training and test datasets, this uncertainty is removed.

If an automated scoring model is fair, as the distribution of student answers which fractions and decimals changes within the training and test dataset (as mentioned earlier, the distribution is the same for both training and test), the model's ability to score them should not change. Again, this is utilizing ABROCA. In simplest terms, the absolute difference between the area under the ROC curves should be minimal between two groups in a prediction task to be considered fair. This shouldn't change given a distribution or more answers which contain decimals or fractions.

To improve the reliability of the results, we re-sample/re-select the test dataset 10 times and evaluate the model's ability to score an open response answer. This form of cross validation allows us to see if the ability to predict the score was only for that unique set of words, or was the performance consistent across multiple iterations.

To summarize, the training dataset is a selection of both answers which contain fractions and decimals (Group A student answers and Group B student answers), using the specified sampling/selection method mentioned above. The test dataset contains the same distribution of data, Group A students, who always use fractions, and Group B students, students who use both decimals and fractions. Again, to reiterate, the balance of Group B in the training and test set are the same. Thus, this can narrow down, if there is a large ABROCA value at different thresholds (more and less decimals/fractions), there is evidence that the fractions and decimals are not impacting the algorithms ability to score the open response answers. If there is a large ABROCA value, there is evidence that there is unfairness in the model's predictions.

As mentioned earlier, the threshold was set to decided whether or not an answer which contains a decimal or fraction is sampled. To reiterate, this is performed by randomly selecting a value between 0 and 1, and if the value falls below the threshold, an answer which contains a decimal is sampled, otherwise, an answer with a fraction in the answer is sampled. We take an incremental approach to the threshold, starting off with a threshold of 0.0, and increasing by 0.10 until a threshold of 1.0 is reached. Again, this allows us to see if there is evidence of unfairness, in terms of ABROCA, at each of the levels. Additionally, if there is evidence, is it occurring with more decimals or fractions?

All of the studies will incorporate a Long Short Term Memory (LSTM) [11] model which utilizes the pre-trained word embeddings to automatically score open response answers. An LSTM model is appropriate here for its sequential attributes. We are able to feed in sequences of words which the LSTM can reference the pre-trained word embeddings to garner semantic meanings of words and the order which they are used. To identify whether or not unfairness is present in an automated open responses scoring model utilizing pre-train word embeddings, we constructed 3 simulated studies with the artificially generated student answers. First, a study which utilizes the similarly balanced simulated training and test datasets, and predicts Group A scores (the group of students who all used fractions within their answers) and Group B (a split of students utilizing fractions and decimals in their answers). Then, we incrementally increase the threshold controlling the split in Group B data (which is similarly controlling Group B threshold in the training set), and utilize ABROCA to directly compare our LSTM's ability to score the student's answers in Group A and Group B separately. We increase the threshold, and repeat. This continues while the threshold increases by 0.1 until a threshold of 1.0 is met. With this, we can gain insights into whether or not the automated scoring of answers of each group is unfair and if so, evidence could be that the use decimals or fractions could be to blame.

The second study looks to identify, if there is variation in our ability to predict scores for Group A and B, whether or not fractions or decimals are the culprit of the potential unfairness in the automated scoring model. In the simplest approach, we remove all non-fraction and non-decimal text from the student answers. Thus, leaving just a fraction or a decimal in the testing dataset. We then develop predictions in the same fashion in the first simulated study mentioned above. This allows us to identify whether or not the pre-trained word embeddings associated with the training data causes the LSTM show unfairness to those using decimals vs. fractions in their answers. Ideally, the ROC curves should be similar. If not, this would suggest that the surrounding 'distractor' words could be influencing the unfairness.

This then leads into the final simulated study. While holding the training and testing of the LSTM constant with the previous two simulated study, this final study replaces the decimals with 'gibberish', or words which are not recognized by GloVe as a pre-trained word embedding. These were chosen by randomly selecting a string of characters. This would increase the amount of 'distractor' words in the text. From this, we can identify whether or not there is unfairness in the LSTM in predicting Group A and Group B scores. If there is unfairness, large ABROCA values, this would suggest that the 'distractor' words are influencing the unfairness. Mainly, the 'gibberish' added does not provide additional information to the LSTM because there aren't pre-trained embeddings associated with those random strings of characters. Thus, a list of 0's is passed through the LSTM and no inferences can be made from those words. Also, since we are only replacing decimals, as the threshold increases, fewer fractions will be available for the LSTM to learn from. So as the fractions drop, so should the ABROCA score. If the ABROCA score increases, there's evidence supporting that unfairness is present from the differing coverage of answer-related tokens within applied methods utilizing pre-trained NLP embedding methods.

From all 3 of these simulated studies, a picture can be painted if the bias present in pre-trained word embeddings causes automated open response scoring models, such as our LSTM, to unfairly grade different groups of students. Similarly, if there is unfairness present, the combination of these 3 studies will allow us to identify what may be causing or influencing the unfairness within our model. Is it the model, is it the embeddings, is the word usage, is the use of decimals vs. fractions? These are the questions which these simulated studies can help to answer.

3.3 Results

First, the results from the standard prediction task of taking the artificially generated student open response answers, in their original form and utilizing pre-trained word embeddings, and utilizing a LSTM to predict what score a student will receive. From our simulated study, Figure 1 presents the ABROCA values at each incremental threshold. Reminder, as the threshold increases, more student answers contain decimals instead of fractions (and vice versa). What

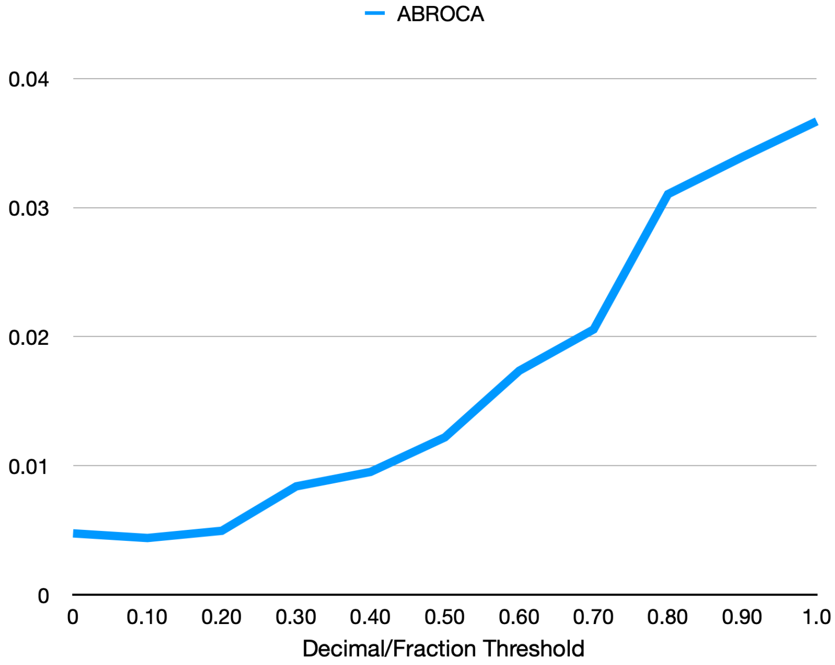

Figure 1: Study 1: ABROCA Values at Incremental Fraction/Decimal Thresholds

is apparent in Figure 1 is that the ABROCA values ever so slightly increase with the more answers which include decimals. However, the amount is almost negligible. Producing ABROCA values near 0 and just under 0.04. This is minimal, that means that the absolute difference between the area under the ROC curves is 0.04. The model appears to able to accurately predict the score a student will receive quite similarly for both groups when decimals, fractions and 'distractor' words are within the answers.

For the second study, which attempts to identify if any unfairness is present when training on answers which contain fractions, decimals and 'distractor' words, but the test dataset only consists of answers with just a fraction or a decimal. This could help to again identify whether or not our ability to score Group A or Group B of students, which use differing levels of fractions and decimals, changes with differing levels of decimals and fractions in the training and testing data. In the end, the ABROCA score was consistently 0 across all thresholds. This meant that no mater the distribution of fractions, decimals and 'distractor' words in the training set, and the distribution of fractions and decimals in the testing dataset, the LSTM with pre-trained word embeddings predicts the score equally for both groups. The LSTM appears to pick up on the rules that answers with 3/4 or 0.75 are correct. Whether decimals or fractions are used doesn't change the LSTM's ability to score different groups with different distributions of fractions and decimals being used.

In the final simulation study, we attempt to identify if the 'distractor' words elicit unfairness in the LSTM utilizing pre-trained word embeddings. By removing all decimals and replacing them with strings of characters that are unidentifiable by GloVe's pre-trained embeddings, we can isolate the model to generating predictions based solely on the surrounding 'distractor' words. Figure 2 shows that the ABROCA score does indeed increase with more unrecognizable words within GloVe's pre-trained word embeddings. When the threshold is set to 0, all the answers contain fractions, and again, we can clearly see that the ABROCA score is quite low, near 0. This is because of the fractions being recognizable to the LSTM. So when attempting to predict scores for Group A and B (B in this case is a split of random characters in place of decimals and fractions), there isn't unfairness present. However, as the fraction wane and disappear, the ABROCA score increases and continues to increase close to 0.18. This may be a bit of a surprise because as the threshold increases, and the number of answers with fractions drops, the LSTM should be able to only identify the similar amounts of words which were randomly selected. In this case, this didn't happen, because Table 1 shows some of the phrases used in the generated student answers were commonly associated with more correct answers. So the LSTM was able to pick up on some of these trends and identify those correlations.

In the end, these simulated studies proved the largest risk for unfairness exists when there is differential coverage of answer-related tokens within applied methods utilizing pre-trained NLP embedding methods. So when answers consist of equally recognizable words within GloVe's pre-trained word embeddings, there's unlikely to be unfairness in the grading. There wasn't evidence that the inherent bias built into the pre-trained word embeddings elicited more unfair scoring of student answers in, in terms of this simulated study. But if there are unbalanced recognizable words and tokens in the student answers, attention needs to be paid to

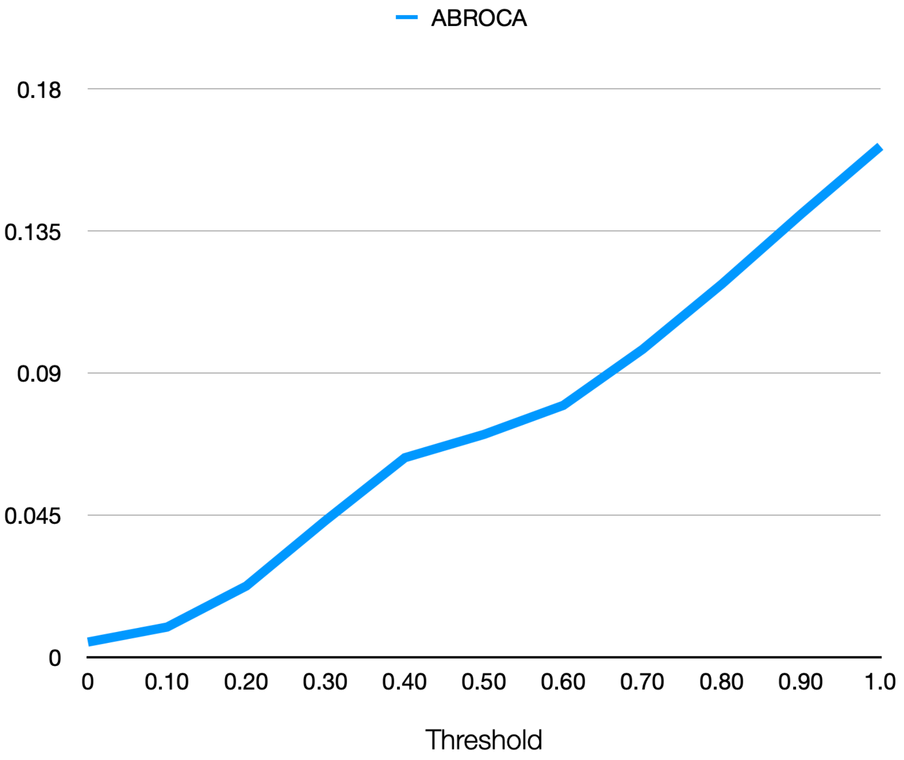

Figure 2: Study 3: ABROCA Values at Incremental Fraction/Decimal Thresholds

potential unfairness in the automated scoring.

4. STUDY 2: MIDDLE SCHOOL MATHEMATICS AUTOMATED SCORING FAIRNESS

While a simulation study is powerful on its own, it is difficult to recreate authentic student data. For the final overall study of this research, we look to once again utilize ABROCA to identify if our own algorithm, trained on genuine student open response answers within ASSISTments, is unfair in its grading of women and men.

4.1 Data

The data consists of two separate datasets consisting of open response questions with associated teacher scores. In its raw state, the dataset consisted of 150,477 total student answers. Within these student answers, 27,199 unique students provided the answers and 970 teachers graded them. These grades and answers span across 2,076 unique problems. It should be noted, that this is the same dataset we used in our study [Blinded for Review]. All of this data comes from middle school mathematics.

However, in its raw state, this data needed filtering down. We make sure to remove any student answers that are empty strings or contained only an image. These filtration steps condensed the dataset down to a total of 141,612 graded student open response answers. In the end, there were a total of 25,069 unique students who answered and 891 teachers graded those answers. After the filtering, there were still 2,042 unique problems attempted. Lastly, the scoring. This was performed on a 5 point scale, where students receiving a 4 is a perfect score.

It should be noted, to be able to perform the fairness analysis using ABROCA, gender was inferred. This performed by cross checking names with the census data. If the name was found only on the women or only on the men's list, it was labeled as such. In any names fell into multiple genders, it was labeled as unknown and excluded from this analysis.

4.2 Methodology and Results

Towards developing our predictions, we utilized another pre-trained algorithm, mentioned earlier, called SBERT. This is a pre-trained sentence embedding algorithm which allowed us to generate a single vector representation of each student answer. We then utilize a Canberra distance to identify which student answers are the most similar. Whichever was the most similar, that was the score we would assign. This approach managed to out do our previous models [Blinded For Review].

While utilizing, once again, ABROCA to identify potential unfairness, we apply this to our algorithm. We were able to show that our SBERT model with Canberra distance manages to fairly score both Male and Female student open response answers. Our model managed an ABROCA of 0.007, which is quite small. Suggesting that our algorithm is indeed scoring Men and Women fairly.

5. LIMITATIONS AND FUTURE WORK

While there were indications of unfairness in cases where there were unbalanced identifiable tokens within the student open response answers, this analysis is strictly middle school mathematics. This type of analysis would need to be applied to additional datasets to get a broader understanding of the potential unfairness in other subjects and age ranges. In terms of our analysis of our SBERT model for scoring student open response answers, while there wasn't unfairness identified, more work needs to be done to explore the embeddings themselves. Pre-trained word embeddings have been shown to have bias built in, but what bias is present in the pre-trained sentence embeddings? This is a question we look to explore further.

6. CONCLUSION

Overall, this study set out to run a simulated study to help identify potential unfairness within models utilizing pre-trained word embeddings. While there is bias present in the embeddings themselves, our simulated study didn't show this bias causing unfair scoring. However, our analysis did show that when developing models with pre-trained embeddings, unfairness can begin to occur when there is an imbalance of recognized tokens in the student answers. More specifically, our simulated study showed that when groups within the data use differing levels of recognized tokens, it increases the chance for unfair scoring.

While our simulated study showed how unfairness can present itself within a scoring model, our model did not show this unfairness. We were able to conduct an analysis of our model with ABROCA to compare our performance scoring Men and Female. In the end, the ABROCA values was nearly 0 at 0.007.

In the end, we were able to utilize a simulated study to help identify potential unfairness in automated scoring models which utilize pre-trained word embeddings. Its been widely noted that those embeddings have bias built in, but out simulated study couldn't show an unfairness in the scoring of differing groups of simulated student answers. Howevwer, this study did show that when student answers have differing levels of tokens recognized, automated scoring models which utilize pre-trained word embeddings can start to unfairly score.

7. ACKNOWLEDGMENTS

BLINDED FOR REVIEW

8. REFERENCES

[1] Y. Attali and J. Burstein. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3), 2006.

[2] T. Bolukbasi, K.-W. Chang, J. Zou, V. Saligrama, and A. Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *arXiv preprint arXiv:1607.06520*, 2016.

[3] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[4] A. T. Corbett, K. R. Koedinger, and J. R. Anderson. Intelligent tutoring systems. In *Handbook of human-computer interaction*, pages 849–874. Elsevier, 1997.

[5] 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.

[6] P. W. Foltz, D. Laham, and T. K. Landauer. Automated essay scoring: Applications to educational technology. In *EdMedia+ innovate learning*, pages 939–944. Association for the Advancement of Computing in Education (AACE), 1999.

[7] J. Gardner, C. Brooks, and R. Baker. Evaluating the fairness of predictive student models through slicing analysis. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 225–234, 2019.

[8] A. C. Graesser, K. VanLehn, C. P. Rosé, P. W. Jordan, and D. Harter. Intelligent tutoring systems with conversational dialogue. *AI magazine*, 22(4):39–39, 2001.

[9] A. C. Graesser, P. Wiemer-Hastings, K. Wiemer-Hastings, D. Harter, T. R. G. Tutoring Research Group, and N. Person. Using latent semantic analysis to evaluate the contributions of students in autotutor. *Interactive learning environments*, 8(2):129–147, 2000.

[10] 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.

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

[12] T. Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, Carnegie-mellon univ pittsburgh pa dept of computer science, 1996.

[13] Z. Kastrati, A. S. Imran, and A. Kurti. Weakly supervised framework for aspect-based sentiment analysis on students' reviews of moocs. *IEEE Access*, 8:106799–106810, 2020.

[14] K. Y. Ku. Assessing students' critical thinking performance: Urging for measurements using multi-response format. *Thinking skills and creativity*, 4(1):70–76, 2009.

[15] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.

[16] J. Liu, Y. Xu, and Y. Zhu. Automated essay scoring based on two-stage learning. *arXiv preprint arXiv:1901.07744*, 2019.

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

[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] A. Onan and M. A. Toçoğlu. Weighted word embeddings and clustering-based identification of question topics in mooc discussion forum posts. *Computer Applications in Engineering Education*,

2020.

[20] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[21] J. Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer, 2003.

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

[23] B. Riordan, A. Horbach, A. Cahill, T. Zesch, and C. M. 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.

[24] J. Roschelle, M. Feng, R. F. Murphy, and C. A. Mason. Online mathematics homework increases student achievement. *AERA Open*, 2(4):2332858416673968, 2016.

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

[26] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, and B. Dolan. A neural network approach to context-sensitive generation of conversational responses. *arXiv preprint arXiv:1506.06714*, 2015.

[27] J. Z. Sukkarieh and J. Blackmore. c-rater: Automatic content scoring for short constructed responses. In *Twenty-Second International FLAIRS Conference*, 2009.

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

[29] K. Taghipour and H. T. Ng. A neural approach to automated essay scoring. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1882–1891, 2016.

[30] K. VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.

Table 2: Training Set Corpus Generation

Answer Length	**Answer Structure**	**Answer Content**	**Fill “1”**	**Fill “2” - Fractions**	**Fill “2” - Decimals**
6	6 - A	i **1** the answer is **2**	‘think’, ‘believe’, ‘feel’, ‘suppose’, ‘guess’	3/4, 1/2, 1/3, 2/5, 3/5	0.75, 0.5, 0.33, 0.40, 0.60
6	6 - B	**1** the answer **2**	‘i arrived at’,‘ i worked out’, ‘ended up with’	3/4, 1/2, 1/3, 2/5, 3/5	0.75, 0.5, 0.33, 0.40, 0.60
5	5 - A	i **1** it is **2**	‘think’, ‘believe’, ‘feel’, ‘suppose’, ‘guess’	3/4, 1/2, 1/3, 2/5, 3/5	0.75, 0.5, 0.33, 0.40, 0.60
5	5 - B	i **1** the answer **2**	‘chose’, ‘thought’, ‘picked	3/4, 1/2, 1/3, 2/5, 3/5	0.75, 0.5, 0.33, 0.40, 0.60
4	4 - A	**1 2**	‘i arrived at’,‘ i worked out’, ‘ended up with’	3/4, 1/2, 1/3, 2/5, 3/5	0.75, 0.5, 0.33, 0.40, 0.60
4	4 - B	my **1 2**	‘guess was’, ‘answer was’, ‘belief was’, ‘answer is’	3/4, 1/2, 1/3, 2/5, 3/5	0.75, 0.5, 0.33, 0.40, 0.60
3	3 - A	i **1 2**	‘chose’, ‘thought’, ‘picked	3/4, 1/2, 1/3, 2/5, 3/5	0.75, 0.5, 0.33, 0.40, 0.60
3	3 - B	it **1 2**	‘was’, ‘is’	3/4, 1/2, 1/3, 2/5, 3/5	0.75, 0.5, 0.33, 0.40, 0.60

Table 3: Test Set Corpus Generation

Answer Length	**Answer Structure**	**Answer Content**	**Fill “1”**	**Fill “2” - Fractions**	Fill “2” - Decimals
6	6 - A	i 1 2	‘arrived at the answer’, ‘thought the answer was’ ‘calculated the answer was’, ‘guessed the answer was’	3/4, 1/2, 1/3, 2/5, 3/5	0.75, 0.5, 0.33, 0.40, 0.60
6	6 - B	2 1	‘is the answer i thought’, ‘was the correct choice here’, ‘is what i guessed right’, ‘was what I arrived at’	3/4, 1/2, 1/3, 2/5, 3/5	0.75, 0.5, 0.33, 0.40, 0.60
5	5 - A	1 2	‘im guessing it was’, ‘my work arrived at’, ‘the answer is clearly’, ‘clearly the answer is’	3/4, 1/2, 1/3, 2/5, 3/5	0.75, 0.5, 0.33, 0.40, 0.60
5	5 - B	2 1	‘was what i guessed’, ‘is what i calculated’ ‘was the clear answer’, ‘is the correct answer’	3/4, 1/2, 1/3, 2/5, 3/5	0.75, 0.5, 0.33, 0.40, 0.60
4	4 - A	1 2	‘my guess is’, ‘my answer is’, ‘my work showed’, ‘my thought is’	3/4, 1/2, 1/3, 2/5, 3/5	0.75, 0.5, 0.33, 0.40, 0.60
4	4 - B	2 1	‘is my choice’, ‘from my work’, ‘is my answer’, ‘is the answer’	3/4, 1/2, 1/3, 2/5, 3/5	0.75, 0.5, 0.33, 0.40, 0.60
3	3 - A	2 1	‘is right’, ‘is correct’, ‘i found’, ‘i thought’	3/4, 1/2, 1/3, 2/5, 3/5	0.75, 0.5, 0.33, 0.40, 0.60
3	3 - B	1 2	‘answer is’, ‘choice was’, ‘i guessed’, ‘i thought’	3/4, 1/2, 1/3, 2/5, 3/5	0.75, 0.5, 0.33, 0.40, 0.60