Received: 14 December 2021 | Revised: 14 December 2022 | Accepted: 20 January 2023

DOI: 10.1111/jcal.12793

ARTICLE

Journal of Computer Assisted Learning WILEY

Leveraging natural language processing to support automated assessment and feedback for student open responses in mathematics

Anthony Botelho[1] | Sami Baral[2] | John A. Erickson[3] | Priyanka Benachamardi[2] | Neil T. Heffernan[2]

[1]College of Education, University of Florida, Gainesville, Florida, USA

[2]Dept. of Computer Science, Worcester Polytechnic Institute, Worcester, Massachusetts, USA

[3]Dept. of Analytics and Information Systems, Western Kentucky University, Bowling Green, Kentucky, USA

Correspondence
Anthony Botelho, College of Education, University of Florida, 1221 SW 5th Ave, Gainesville, 32601, FL, USA.
Email: a.botelho@ufl.edu

Funding information
Institute of Education Sciences, Grant/Award Numbers: R305A120125, R305A170137, R305A170243, R305A180401; National Science Foundation, Grant/Award Numbers: 1109483, 1252297, 1316736, 1440753, 1535428, 1636782, 1724889, 1759229, 1822830, 1903304, 1917713, 1917808, 1931523, 1940236, DRL-1031398; Office of Naval Research, Grant/Award Number: N00014-18-1-2768; Schmidt Futures; U.S. Department of Education, Grant/Award Numbers: P200A180088 P200A150306, R305A120125 R305A180401 R305C100024, R305A170137 R305A170243 R305A180401

Abstract

Background: Teachers often rely on the use of open-ended questions to assess students' conceptual understanding of assigned content. Particularly in the context of mathematics; teachers use these types of questions to gain insight into the processes and strategies adopted by students in solving mathematical problems beyond what is possible through more close-ended problem types. While these types of problems are valuable to teachers, the variation in student responses to these questions makes it difficult, and time-consuming, to evaluate and provide directed feedback. It is a well-studied concept that feedback, both in terms of a numeric score but more importantly in the form of teacher-authored comments, can help guide students as to how to improve, leading to increased learning. It is for this reason that teachers need better support not only for assessing students' work but also in providing meaningful and directed feedback to students.

Objectives: In this paper, we seek to develop, evaluate, and examine machine learning models that support automated open response assessment and feedback.

Methods: We build upon the prior research in the automatic assessment of student responses to open-ended problems and introduce a novel approach that leverages student log data combined with machine learning and natural language processing methods. Utilizing sentence-level semantic representations of student responses to open-ended questions, we propose a collaborative filtering-based approach to both predict student scores as well as recommend appropriate feedback messages for teachers to send to their students.

Results and Conclusion: We find that our method outperforms previously published benchmarks across three different metrics for the task of predicting student performance. Through an error analysis, we identify several areas where future works may be able to improve upon our approach.

KEYWORDS
automated assessment, feedback recommendation, natural language processing, open responses, sentence embeddings, similarity

1 | INTRODUCTION

Educational technologies strive to support educators and students in ways that augment instructional practices. These technologies, for example, can help an instructor monitor student performance at levels of granularity and at scales that are infeasible or impractical in more traditional classroom environments. Advancements in machine learning and artificial intelligence have enabled educational technologies to aid in teachers' decision-making processes; with access to large sums of current and historic educational data, these technologies show promise in their capacity to suggest effective actions that teachers can take to support their students' learning.

These systems are developed to support both teachers and students through functions and tools that utilize collected data in various ways. These functions include scaffolding problems (Ringenberg & VanLehn, 2006), worked examples (Roll et al., 2011), or even answer-specific feedback to help remedy known common errors (Selent & Heffernan, 2014). Even simple correctness feedback provided to students can be beneficial for their learning (Kehrer et al., 2013), and is one of the most prominent benefits of learning systems. In considering these supports, there still remain several challenges in addressing all types of problems that teachers commonly assign to students.

Common across domains, though particularly in the domain of mathematics on which this paper is primarily focused, there are two types of question: close-ended and open-ended. Close-ended problems have a single or finite number of accepted answers (for example a multiple-choice question or fill-in questions) that are easily recognizable by a system through a simple matching approach; student answers can be compared, character by character, to a set of recognized answers for a direct match to determine correctness. Support learning systems is generally confined to these close-ended problem types. For instance, it is easy for a system to understand that the accepted value for x in the equation $x + 4 = 8$ is 4. If a student were to answers with the value of 12, the system may easily be able to provide the correctness score with a feedback message that highlights the student's mistake. Open-ended questions, however allow students to express their understanding of concepts through natural language (e.g., Figure 1); while there are still a finite number of conceptually acceptable answers, such responses can vary greatly making it impossible to utilize a direct matching approach as can be used for close-ended problems. In many K-12 mathematics classrooms, these types of open-ended questions are used by teachers to assess their student's understanding and thought process about the assigned topic.

The challenges posed by open-ended problems in developing tools to better support the provision of various forms of feedback become even more apparent in mathematics-prevalent domains. Distinctive even from other contexts pertaining to language arts, student answers to open-ended questions in mathematics often include a combination of natural language with other artefacts including images, tables, mathematical symbols, as well as equations and expressions. While the number of words in a language such as English is large, it is finite, making it easier to represent words (or sequences of commonly-occurring words) as distinctive components when building automated systems to process natural language. The inclusion of numbers, as an infinite set, complicates traditional approaches to natural language processing, especially when appearing within more complex expressions.

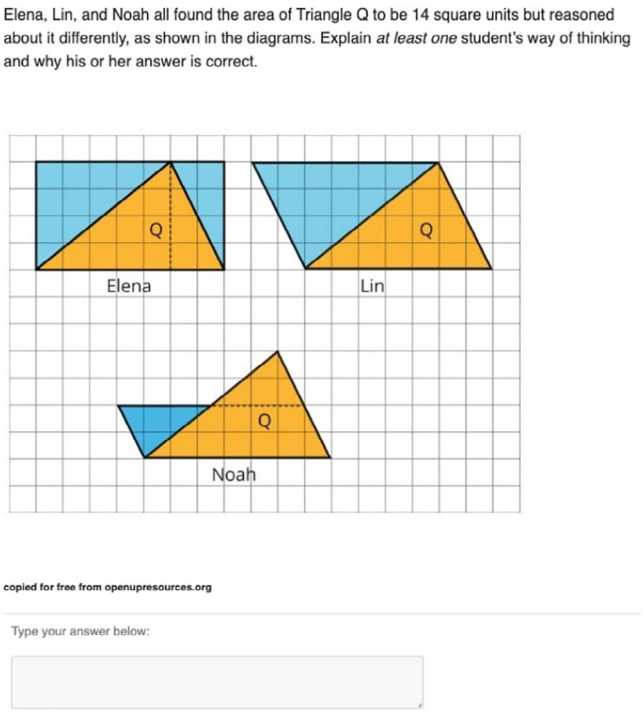

FIGURE 1 Example of an open-ended question taken from openupresources.org

1.1 | Research questions

In this paper,[1] we provide a deeper examination into the methods and framework underlying recent work in the space of automating assessment in the domain of mathematics (Baral et al., 2021). We introduce, and develop, an approach which utilizes sentence level semantic representation of student open responses through Sentence-BERT (SBERT). With the ultimate goal of helping teachers assess and provide feedback to open-ended problems in mathematics through the development of an automated assessment system, this work explores the following overarching research questions:

1. How can historic student data be leveraged in assessing and suggesting feedback for student answers to open-ended problems in mathematics?
2. When examining student open-response answers, which similarity metrics most closely align with how a teacher may group similar student answers?

[1]It is important to acknowledge that this work expands upon previously published research conducted in this space (Baral et al., 2021).

3. What are the characteristics of student work that impact the performance of our proposed methods?

In addressing our research questions, we identify four contributions presented in this work. First, we conduct an empirical analysis to compare different feedback recommendation policies utilizing traditional and state-of-the-art NLP and machine learning methods. Focusing on developing methods to suggest feedback as well as assessment, we develop a data-driven metric and procedure to evaluate automated feedback recommendation policies in an offline manner based on how teachers identify similar student answers. We then introduce a method to provide automated scores and feedback suggestions to teachers to give to their students' open-ended work based on the similarity of student responses. Finally, to explore what impacts the performance of our methods, we conduct an error analysis to investigate the answer-level characteristics that correlate with the magnitude of error observed in our proposed methods.

2 | BACKGROUND

Intelligent tutoring systems (ITS) and computer-based learning platforms have been utilized by teachers and students for several decades (c.f. Corbett et al., 1997), but their adoption has increased in recent years. With the rise of COVID-19, teachers' reliance on these online learning platforms such as ASSISTments (Heffernan & Heffernan, 2014), McGraw Hill's ALEKS™ and others have grown. Studies conducted to evaluate online learning platforms, have identified promising results that point toward their success in helping to improve student learning over more traditional instructional approaches. In the case of ASSISTments, as it is the platform from which the data used in this work was collected, such an efficacy trial found that use of the platform nearly doubled student learning over the course of a year (Roschelle et al., 2016); at least part of this outcome has been attributed to the simple offering of immediate feedback, as previous studies have identified (Kehrer et al., 2013). Similarly, the study from (VanLehn, 2011) highlights the positive effects of intelligent tutoring systems on student learning, and that these intelligent tutoring systems have the opportunity to be almost as effective as human tutors. Most learning platforms, particularly in the context of mathematics, largely focus on closed-ended problem types as they are well-structured and are easier to automatically grade. As these types of problems are easier to automate, there has been a considerable amount of work and research in improving feedback to these types of questions. However, Ku (2009) discussed that providing only one question type, such as multiple-choice, would be inadequate in capturing the students' rationale or process of thinking. Similarly, Chi et al. (1994) highlights that pedagogically engaging students in explanation about their works and making arguments and justifications about their understanding of the concept, leads to the development of conceptual understanding and finally better learning. Similarly, in comparing close-ended questions (such as multiple-choice) and open-ended questions, Martinez (1999) has identified and examined the different levels of cognition required for each question type, suggesting that open-ended questions require a wider spectrum of cognition than that of a close-ended question. A similar study by Kramarski and Zeichner (2001) highlights the importance of metacognitive feedback on questions that asks for the understanding of the approach has better impact on math learning than just a result based feedback on result problems.

While past studies have suggested the importance of open-ended questions, there is still only limited support in most learning platforms for this type of problem; many systems do not even support this problem type in the context of mathematics. As Buckles and Siegfried (2006) highlights, there are certain affordances in omitting open-ended problems as they are considerably more difficult to assess with support from automation. Recognizing the benefits of these problems, however, efforts are being made to automate the assessment of open-ended answers (c.f. Attali & Burstein, 2006; Foltz et al., 1999; Zhao et al., 2017; Erickson et al., 2020). In this work, we present an approach to help provide support for not only the assessment of open-ended problems, but also to help aid in the writing of feedback in support of student learning.

2.1 | Natural language processing

To address the problem of automatically assessing student answers to open-ended problems, such approaches need to be able to quantify aspects of student responses. In most modern approaches, this is usually accomplished through the application of natural language processing (NLP) methodologies. While the study and application of NLP broadly encompasses numerous aspects of the study of language, for the purpose of this work we refer to NLP in terms of the narrow scope of methodologies used to convert language into numeric representations; the primary goal of most modern methods, as described further below, is to represent language in such a way that describes syntactic and semantic features.

Within this, there has been a long history of research pertaining to automated short-answer grading (ASAG), following several trends distinguishing scoring for what we would refer to as close-ended problems and more open-ended formats (see, for example, Burrows et al., 2015). Within this work, the authors also highlight several distinctions between ASAG and the scoring of longer essay responses (automated essay grading, or AEG). Our specific context is characterized by open-ended responses that consist of, at most, a couple of sentences. Across these areas of automatic scoring of language, Burrows et al. (2015) identifies sets of methods used to approach this problem as following "eras" ranging from concept mapping (e.g. C-Rater; Leacock & Chodorow, 2003) to machine learning approaches discussed later in this section.

There have been many methods proposed in the past as to how to best represent text in a manner that captures syntactic as well as semantic meaning. The simplest way to represent language is perhaps with a bag of words approach. By adding up the number of times the word occurs, that can be the number which represents said word.

While this has been the foundation of recent studies (e.g. Graesser et al., 2000; Sordoni et al., 2015), the bag of words approach is often utilized as a baseline of comparison for more complex methods. Similarly, some research utilizes n-gram approaches for text classification (Cavnar & Trenkle, 1994). In such a method, instead of having a list of individual words, a list of grouped words could be counted (e.g. perhaps observing two- or three-word sequences as a single grouping to form a bi-gram and tri-gram representation, respectively); such a method helps to capture the context of particular words by observing how each word is situated in relation to others. Not only has this been used in interpreting speech or text, it has also been utilized in other contexts, such as in detecting malicious code (Abou-Assaleh et al., 2004). Other common foundational methods attempt to measure the importance of words based on the frequency or scarcity of their usage; term frequency inverse document frequency (TF-IDF), for example, was developed to provide a weighting measure designed to discount overly-common words and focus on keywords that help to provide better meaning (Ramos, 2003). A recent study showed success in automatically grading student open-ended questions using TF-IDF (Erickson et al., 2020).

What is missing from a bag-of-words-based approaches is any relational or contextual understanding of the words; even considering n-gram methods, though context is represented through adjacent words, such methods are unable to measure deeper relationships between the ideas present in the given text. These foundational methods provide some measure of word frequency (and perhaps importance), but it fails to provide and relational information; we argue that both of these characteristics are important in considering the assessment of student open-ended work. With recent advancements in deep learning, researchers have developed approaches that attempt to embed contextual information within high-dimensional language representations; two such notable methods are those of Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). Though varying in the specific formulation, these deep-learning-based embedding methods attempt to project words into an embedding space such that the semantic relationship between words is maintained in their distance in that space; if the two vectors are far apart within the vector space, it is presumed that they are less likely to be related or similar. An additional benefit to such embedding methods is the ability to "export" the learned representations so that they can be utilized in broader applications; embeddings can be generated on large corpora of language data and then be used in various contexts as a pre-trained representation, supporting applications where such a robust corpus may be difficult to collect (and also such pre-trained methods add statistical power in that the representations can incorporate semantic and contextual information from a large and diverse sample set).

Expanding from the development of word embeddings, higher representations at the sentence- and document-level have also emerged. Instead of developing a vector representation of each individual word, approaches such as Doc2Vec (Le & Mikolov, 2014), aim to generate a single embedding to represent an entire document or multiple paragraphs. Likewise, other approaches such as the Universal Sentence Encoder (Cer et al., 2018) and Sentence-BERT (Reimers & Gurevych, 2019) have gained popularity in their ability to represent sentences as a single vector, offering opportunities to use such representations to capture the relationships between broader ideas that may be split across words, clauses, sentences, and paragraphs.

2.2 | Auto assessment of open-ended problems

There has been a growing body of research around the automated assessments of open-ended response in conjunction with the emergence of improved NLP methodologies. Prior works on automated assessment of student open-responses are of ranging complexities that are based on the type of the answer text and the subject domains (Burrows et al., 2015). Such works present various automated methods to help teachers assess short answers and essays in several domains (Basu et al., 2013; Brooks et al., 2014; Goularte et al., 2019; Leacock & Chodorow, 2003; Sultan et al., 2016; Zhao et al., 2017). Studies such as Basu et al. (2013) and Brooks et al. (2014) have implemented various clustering based techniques to grade short textual answers. C-rater (Leacock & Chodorow, 2003) used grading rubrics and the decomposition of scores into multiple knowledge components to assess the correctness of short answer questions. Study from Sultan et al. (2016) proposes methods on short answer grading tasks based on the semantic similarity of the student response with the correct response. Other more recent works (e.g. Riordan et al., 2017; Zhao et al., 2017) have been based on various deep learning methods to assess open-ended answers. While these works have mostly been applied in non-mathematical domains, there are other works such as Lan et al. (2015) which focuses on the auto-assessment of open-ended questions in mathematics, emphasizing the unique challenges present in representing mathematical language and expressions. Subsequent work in the context of mathematics (i.e. Erickson et al., 2020) discusses challenges in developing auto-scoring models for open-ended questions in such a context. In their work, they offer a comparison of various models utilizing machine learning (e.g. random forest and XGBoost; Chen & Guestrin, 2016) and more complex deep learning (e.g. Long Short Term Memory (LSTM) networks; Hochreiter & Schmidhuber, 1997) techniques; they combined these with NLP methods to automatically score open-ended responses.

Beyond correctness feedback, there are also some works that have explored the generation of other forms of feedback for natural language contexts. Recent work, for example, has developed and evaluated a discourse-based feedback system that communicates with students through a chat-like interface (Grenander et al., 2021). Other prior research has explored the delivery of feedback through more structured scaffolding (Grossman et al., 2019).

2.3 | Developing QUICK-Comments tool

Building upon these prior works, the studies on which we report in this work follow the development and pilot-testing of an automated assessment and feedback recommendation tool called QUICK-Comments Tool. The design of this tool draws inspiration from

Google's SmartReply (Kannan et al., 2016). This tool, and others like it that have since been developed, have become widely-used to help users respond to email and other forms of textual communication. Like SmartReply, the goal of QUICK-Comments Tool is to provide teachers with three suggested feedback messages to provide for each student answer to an open-ended math problem; in addition, the tool provides a suggested assessment score for each response, allow teachers to utilize these suggestions or ignore them to formulate their own scores and feedback for students.

These technologies often rely on their ability to effectively compare new experiences with historic data. If a particular observed scenario has been seen in the past, it is likely that a course of action that was previously successful in such a case may be appropriate in the present as well. In the case of SmartReply, email responses may be re-used in other contexts (consider how certain replies such as "sounds good" or "thanks!" are appropriate for a wide range of contexts). Expanding this example to the context of education, historic contexts are often used to inform how to approach similar scenarios in the future. If a student under-performed in mathematics classes in high school, for example, it may be appropriate to recommend that the student enrols in a remedial math course in college based on similar students benefiting from such a selection in previous years. In such a practice, however, the success of the recommendation is based on the system's ability to compare and quantify similarities between two artefacts (i.e. the system needs to be able to find historic examples that are similar to what is currently being observed). In the course recommendation example, the system must quantify and compare student performance in high school mathematics courses using, for example, letter grades or students' grade point averages.

Tools like SmartReply typically utilize natural language processing along with several machine learning methods such as LSTM deep learning networks (Hochreiter & Schmidhuber, 1997) to process users' email and then recommend appropriate responses. While the original SmartReply paper describes a generative approach (where responses are being generated word-for-word), many similar recommendation systems instead rely on a case-selection method (i.e. from a pool of known possible artefacts, select the one that best applies). In either case, however, this technology, like other recommendation systems, relies on the method's ability to identify similar known examples in order to make informed recommendations as to how to proceed.

3 | DEFINING SIMILARITY

As the concept of "similarity" is a prominent aspect of this current work, it is important to discuss our definition of this term as it aligns with different measures of distance and relatedness.

To illustrate the concept of similarity, consider the example illustrated by Figure 2. Which of the three objects, A, B, or C, is most similar to the target object in the upper left? Is it possible to order these artefacts from most similar to least similar? Ignoring context, it is not likely that readers would unanimously agree on the answers to these questions due to the number of dimensions in which the artefacts can be compared. Similarity here can be expressed in regard to shape, rotation, colour, or any number of other attributes. Each artefact exhibits similarities and differences along each of these dimensions; without more information (or more structure to the problem) it is impossible to know which dimensions should be given higher importance. In other words, the difficulty of this task stems from the combination of the dimensionality of the artefacts and the unknown weights of these dimensions for comparison.

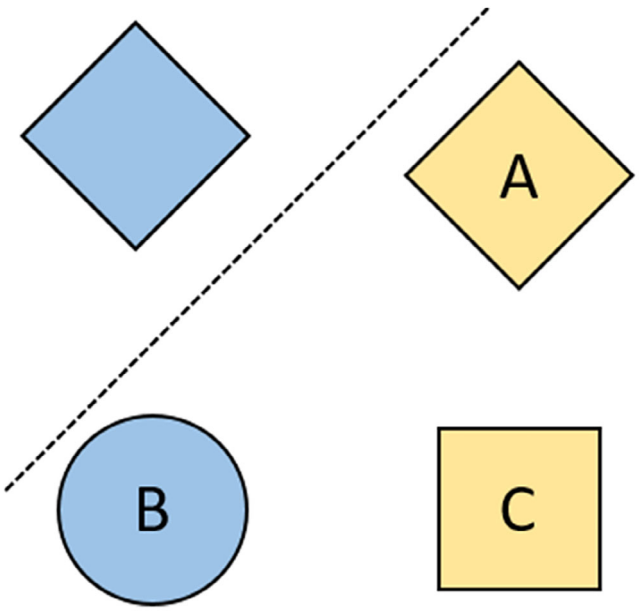

FIGURE 2 Example of how similarity can be defined along multiple dimensions of comparison.

In a practical sense, this challenge is even greater when even the artefacts are difficult to describe, such as is the case with natural language. Consider, for example, the sentences "see Spot run" and "Spot runs fast" for comparison. In what ways are these sentences similar? Semantically, they both refer to "Spot" and describe Spot's action of running, but there are many other ways to compare these. Both of these are similar in their count of the letter "s," both use the same number of spaces and have the same number of words. Likewise, apparent similarities could be viewed as differences; the word "Spot" appears earlier in the second sentence. In this way, there are multiple "correct" ways of measuring the similarity of these sentences; it is just likely the case, however, that some methods of measuring similarity are more useful than others depending on the context.

In the most abstract sense, similarity can be defined as the distance between quantified artefact representations. Once an artefact can be quantified along its various attribute dimensions, these values then represent the artefact's point within a representation space. Measuring the distance between these points reveals the similarity of the given artefacts.

4 | LEVERAGING HISTORIC DATA

In this work, we propose a method, or framework, for leveraging historic student responses for the tasks of automated assessment and

feedback recommendation based on the idea of collaborative filtering (Su & Khoshgoftaar, 2009). Collaborative filtering is a common method utilized by recommendation systems based on the idea of removing irrelevant suggestions to better focus on relevant relationships in data. Distinctive from other works that have approached this problem from a supervised learning perspective, we instead re-envision the task as a similarity ranking problem that intends to utilize the large amount of historic data that is collected through educational technologies. The reason for this is due to the nature of open-ended problems, where the sample space is too expansive and sparse to effectively evaluate supervised approaches using traditional methods. For example, if I wanted to predict an appropriate feedback message to give to a student, this is often a near one-to-one relationship between student answers and feedback messages (i.e. it is rare that a teacher says exactly the same thing for two different student answers); as the label space is nearly just as large as the sample space, it would be nearly impossible to train (and evaluate) a machine learning model in a traditional supervised manner. Instead, the problem can be re-framed to identify groups of student work for which feedback for one member is appropriate for all members, in which case the challenge becomes developing methods that accurately identify group membership as represented by a binary outcome (does a student sample belong to a given group or not) or as a continuous similarity measure (e.g. a likelihood that two student samples belong to the same group for all pair-wise comparisons). Thus, if we are presented with a reasonable way of identifying "ground truth" groupings by, for example, letting teachers define categories of student work, we can evaluate measures of similarity (and other estimation methods) in terms of how well they agree with teacher-provided groupings. In the remainder of this section, we describe the framework proposed, with the next section then describing an empirical study used to evaluate this approach with data collected from teachers.

To begin, let us assume that, for a given problem or context P_0, we have a list of historic samples $A_{0...n-1}$. For a new student sample, A_n, we want to rank samples $A_{0...n-1}$ in regard to their measured similarity to A_n. In order to achieve this, we must project all samples $A_{0...n}$ onto a feature space using one of a set of representation methods $R_{0...j}()$, such that answers are characterized as comparable feature vectors $X_{0...n}$. For every pair-wise comparison of X_n to $X_{0...n-1}$, one of a set of similarity-measuring methods $S_{0...k}()$ are applied to generate a set of scalar distance values $D_{0...n-1}$ that measure how close, in terms of the representation space, the new sample is compared to each of the historic samples. Finally, $D_{0...n-1}$ can then be sorted to identify the ranking.

It is important to note that we present this framework in such notation as to be agnostic to the type of student data observed. While we present this work in the context of student answers to open-ended problems, we posit that this framework could extend to compare other types of data as well including, for example, student process data to identify similar sequences of student interactions.

We must identify some assumptions that must be met in practical application of this framework. First, given a particular problem or context of interest, we assume that this context has a sufficient number of student samples and that those student samples have been previously assessed by a trained assessor. In terms of our application to student open-ended work, we must assume that there is a sizable number of historic answers that have teacher-provided scores and feedback messages so that we may recommend these for a newly-collected student response.

The second assumption we make is that there is a set of historic samples that belong to the same conceptual grouping as a newly-observed sample, in a binary sense. This may be unreasonable, as just because a particular historic sample is the "closest" to a given new sample within the representation space, does not necessarily mean that it should belong to the same conceptual grouping. For this reason, it is recommended that, in practical application, the framework be expanded with a determined threshold $T_{0...k}$ which acts as a cut-off for considering two samples as similar or not Such a threshold could then, for example, help to account for scenarios where a new student sample is completely unlike anything previously recorded in the set of historic samples. In either case, this risk may be further mitigated through a human-in-the-loop approach, where a trained assessor (i.e. teacher) can simply choose to ignore recommendations that are inappropriate before they are given to a student.

A third assumption here is that we have an appropriate representation method R and similarity procedure S that collectively produce a meaningful distance value for each comparison. While this is perhaps the largest assumption, it is also one that can be tested and evaluated. In reality, we have access to a large number of procedures, $R_{0...j}$ and $S_{0...k}$, collectively representing potential "recommendation policies" to measure similarity. In comparing different policies, however, we need a ground-truth value of similarity as defined by teachers with which we can compare. The method by which we calculate this value is described in the next section.

5 | STUDY 1: COMPARING MEASURES OF SIMILARITY

While this framework provides a quantitative ranking of historic samples based on a likelihood of each pair-wise set of samples $A_{n,0...n-1}$ belonging to the same conceptual group, we must define $R()$ and $S()$, as previously stated. As we are observing answers to open-ended problems, we can compare existing NLP representation methods in conjunction with existing measures of similarity to evaluate which combinations most closely align with teachers' definitions of similarity.

In order to evaluate recommendation polices, there are potential online and offline approaches that can be utilized. To evaluate the policies in an online sense, we could simply build the proposed system, and compare the effectiveness of policies based on how often the recommendations are chosen by teachers. There are of course several issues with this method in that it could take a long time to evaluate a large number of policies. Ideally, we would want to use an offline method, effectively simulating or approximating teachers' choices of recommendations; in this way, a large number of policies can be

evaluated simultaneously using a common dataset. While only an approximation of how teachers would utilize recommendations, offline methods are often used to first filter the number of likely-optimal policies to a small number of candidates that are then further evaluated in an online manner.

5.1 | Evaluating recommendation policies

In this work, we evaluate the proposed policy in an offline manner using a dataset constructed through close collaboration with a cohort of 17 teachers from across the United States. The goal in constructing this dataset was to develop a measure representing similarity as defined by the group of teachers as a whole. Having such a measure provides a ground-truth value of similarity with which we can compare our recommendation policy distance values.

The data was constructed by first sampling student answers to open-ended problems from widely-assigned open educational resources (OER) in the context of middle school mathematics. We then randomized sets of student answers from the same problem and presented them to subsets of the teachers. With these responses, per problem, we asked the teachers to group the responses into any number of desired categories. We gave no further instruction regarding how to group the student answers nor the number of categories to use. In this way, the teachers could decide, through heuristics or inherent processes, how to identify "similar" answers by placing them within the same abstract category. Not only this, but since multiple teachers performed this categorization for the same set of responses and problems, we also are able to capture variation in how teachers define and identify similarity. There were initially 78 distinct open-ended problems with sampled student answers, but was ultimately filtered to 67 problems due to some problems having been categorized by fewer than two teachers. As a final filtering step, empty student responses were also dropped from the dataset. After all filtering, there were a total of 5,539 student answers across 67 problems. A sample of these responses and their associated teacher response categories are presented in Table 1.

From Tables 1, 3 separate student responses are presented from the same problem. In this case, correct answers tend to include the value 34.5. Within the table you can see for this problem, Teacher 1 has used the category "C" when a student provided a correct answer. Therefore we can infer student answers with the category "C", the teacher considered similar, and would elicit a similar response. Within our data, teachers created an average of 5.53 (SD = 1.93) distinct student answer categories across all problems, with a median of 5 categories (min = 2, max = 18).

With this data, we constructed a metric which we call the Teacher Agreement Score (TAS). This value is calculated for a given recommendation policy by first applying the proposed recommendation method presented in Section 4 to generate the top R most-similar responses (where $R = 3$ in our particular evaluation) from a selected holdout answer as A_n. From these selected answers, the sample-level TAS is calculated as follows:

TABLE 1 Sample student answers for a single problem and associated teacher response categories

Student answer	Each Teacher's category: Category T1, T2, T3
I divided 7.5 by 0.75 and got 10 then I did 10 times 3.45 and got 34.5.	C, B, K
I divided 3/4 by 3.45 and got 0.21 then i multiplied 3.45 by 0.21 until i got 7.5	I, D, J
I know this because I divided 3.45 by 3/4 and got 4.6 and times 4.6 by 7.5 and got 34.5	C, A, K
I did 3.45 divided by 0.75. I got 0.75 because 3 divided by 4 is 0.75. When I divided 3.45 and 0.75, I got 4.6. I then did 4.6 multiplied by 7.5 and got 34.5.	C, A, K

$$TAS_i = \frac{1}{R}\sum_{j=0}^{R}\frac{1}{T}\sum_{t=0}^{T}\begin{cases}1, \text{if } C_{i,t} = C_{j,t}\\ 0, \text{Otherwise}\end{cases}$$

This equation calculates TAS for holdout sample i by comparing the teacher-given categories of this response in comparison to the categories provided for the selected R responses for all teachers T with provided categories. In other words, for each pairwise comparison of sample i to samples in R, the metric simply counts the number of teachers that agree that the two belong to the same category, averaging this over all teachers and response pairings. This process is repeated in a hold-one-out manner (observing each student answer as the selected holdout) and an average TAS is calculated for the observed policy in regard to the given problem. Finally, this process is repeated across all 67 problems and an average and per-problem TAS is used to compare each policy.

To give an example of this calculation, consider Table 1. If the last row was used as a holdout sample and the first 3 rows were the identified 3 most-similar responses, the calculated TAS_i for this sample would be 0.556. This is, again, calculated by comparing for matching categories within each teacher for each response; the categories match for 2/3 teachers when comparing to the first row, 0/3 for the second row, and 3/3 when comparing to the third row. These values are then simply averaged to find the 0.556 value. The process would then continue by rotating the holdout sample.

Ultimately, a TAS close to 1 suggests that the observed policy agrees with how teachers would define similar student responses. To clarify, TAS represents a percentage of teachers that would agree with a method that identifies sets of similar answers. In this way, policies exhibiting higher scores are, in theory, more likely to be utilized by teachers.

5.2 | Evaluating policies

Now that we have defined both our proposed method for recommending feedback and our evaluation method derived from real

data, we present an empirical analysis to both exemplify these methods as well as compare several potential recommendation policies of varying complexity. These methods are further detailed in the sections below.

5.2.1 | Universal sentence encoder

As introduced in the Background Section, several NLP methods of representing text have grown in popularity for their ability to capture the semantic meaning of not only words, but also full sentences and even paragraphs. The first method that we explore within our empirical analysis is the Universal Sentence Encoder (USE; Cer et al., 2018). While other NLP methods often build numeric representations of individual words, the USE builds a single vector representation for a given sequence of words within a high-dimensional vector.

Once a sentence-level embedding is generated for each response, a distance measure (described below) can be applied to measure the "closeness" of other student answers in vector-space. As this method is meant to capture the semantic meaning of the sentence, and leverages complex deep learning methods to do so, this method has the potential to allow for comparisons beyond the surface-level features of the text.

5.2.2 | Sentence-BERT

Developed even more recently and arguably considered to be the current state-of-the-art of sentence representation is the second method of comparison: Sentence-BERT (Reimers & Gurevych, 2019). Developed from the word-level representation method of BERT, this method constructs a high dimensional vector representation of sentence- or paragraph-level text similar to that of the Universal Sentence Encoder. This method, however, is based on what is known as a "siamese network" architecture. This type of network attempts to incorporate textual and semantic similarity into the generated embeddings. In this way, this method represents the most complex of representation methods compared in the current analysis.

Throughout all our analyses, we utilize a pre-trained version of Sentence-BERT. This model has been trained on a large corpus of samples collected from Wikipedia. It is important to note that, while prior research has explored ways in which pre-trained models such as these may be "fine-tuned" or trained on context-specific data (Shen et al., 2021), no parameter tuning was applied in any of the analyses described in this work.

5.2.3 | Levenshtein ratio

Among the simplest methods of comparing the likeness of two samples of text is that of Levenshtein Distance. This approach examines strings of characters and calculates a distance based on how many need to be changed to turn one string into its comparison string. For example, if a student A said 'the answer is 45' and student B submitted an answer with 'the answer is 46', the distance would be 1. However, if student B answered with 'I think the answer is 46', the distance would be 9. Clearly, there are disadvantages to this approach, mainly the distance could be larger between two answers, but their content is the same. However, when considering a character level distance metric, could this out perform more modern approaches? For the purposes of the paper, we utilize the Levenshtein Ratio which calculates the distance and converts it to a similarity ratio which is meant to account for the comparison of strings of different lengths.

This method acts as a baseline comparison method due to the simplicity of the approach. However, it is likely that surface-features of text (i.e. the use of particular keywords within student answers) may actually prove to be a highly-weighted attribute among the teacher comparisons.

5.2.4 | Distance metrics for similarity

While the above methods generate representations of student answers, the method of calculating the distance between representations is still needed. In this regard, we observe three different methods within this analysis: Euclidean Distance, Cosine Similarity, and Canberra Distance; these were chosen both for their prominent usage in previous NLP research and also for their notable differences in meaning. As described in an earlier section, Euclidean Distance observes the magnitude of the geometric distance between two vectors while Cosine similarity observes the difference in angles produced by two representation vectors. Canberra Distance, while not as widely known as the other two, has been applied in areas of computer science as a means of comparing ranked lists (Jurman et al., 2009). Each of these distance measures are applied to the above representation methods (excluding the Levenshtein Ratio) and the TAS measure is calculated for each as described in Section 5.1.

6 | STUDY 1 RESULTS

As mentioned earlier, after all the filtering, there were 67 total problems with 5,539 student answers. From this, we calculated the overall Teacher Agreement Scores for each approach and distance measure as shown in Table 2. In comparing representations, Sentence-BERT appears to outperform the other methods to a statistically significant degree, as observed by the non-overlapping 95% confidence bounds; while statistical significance and overlapping/non-overlapping confidence bounds are not necessary to compare performance differences, they provide evidence in this case that Sentence-BERT may provide representations that better align with teachers' implicit heuristics. While USE exhibits a higher average TAS than the raw character representation across distance metrics, the differences are not found to be statistically significant. Observing the TAS metric of all of these approaches, however, even the lowest-performing approach (i.e. raw character representation with Levenshtein Ratio) exhibits an

TABLE 2 Overall teacher agreement scores

Representation (R)	Distance metric (S)	Average TAS	95% Confidence bounds
N/A	Levenshtein ratio	0.536	[0.510, 0.562]
Universal sentence encoder	Euclidean	0.556	[0.530, 0.582]
Universal sentence encoder	Canberra	0.554	[0.527, 0.581]
Universal sentence encoder	Cosine	0.556	[0.530, 0.582]
Sentence-BERT	Euclidean	0.621	[0.596, 0.646]
Sentence-BERT	Canberra	0.623	[0.598, 0.648]
Sentence-BERT	Cosine	0.623	[0.598, 0.648]

TABLE 3 Teacher agreement scores per problem. It should be noted that the'Number of Times Best Teacher Agreement Score for Problem' sums to over 76/67; this occurs because there were 9 cases where two approaches scored the same score for that problem. Thus, either of the approaches would be considered acceptable

Representation (R)	Distance metric (S)	% best teacher agreement score	Number of times best teacher agreement score for problem
N/A	Levenshtein	1.492	1/67
Universal sentence encoder	Euclidean	11.94	8/67
Universal sentence encoder	Cosine	11.94	8/67
Universal sentence encoder	Canberra	4.48	3/67
Sentence-BERT	Euclidean	17.91	12/67
Sentence-BERT	Cosine	25.37	17/67
Sentence-BERT	Canberra	40.30	27/67

agreement score that indicates greater than 50% agreement with teachers; in other words, even in the worst-performing case, we see that the method utilizing Levenshtein Ratio identifies pairs of student responses that 53.6% of teachers would agree to represent similar student answers.

Overall, the strongest performing approach, in terms of Teacher Agreement Scores, was Sentence-BERT. Consistently, Sentence-BERT managed the highest average Teacher Agreement Score across all the problems with all distance measures. However, there were only small differences in performance observed across distance metrics paired with the Sentence-BERT representation. Of these metrics, Canberra and Cosine similarity results in the highest observed average TAS of 0.623; this suggests that an estimated 62.3% of teachers would agree with the sets of similar answers identified by these methods. While the highest performing among methods examined in this work, a 62.3% agreement leaves a large margin for improvement and suggests that teachers are considering several dimensions of comparison that are seemingly missed by our observed policies.

What is evident is that different combinations of representations and similarity methods varies in their ability to identify suitable similar student answers, as seen in Table 2. While this exhibits our ability to evaluate our similarity calculations, and could be scaled to apply to new answers within the same problem, we set out to see how the models performed not just overall, but on a per problem basis.

Table 3 provides a breakdown of the performance of each combination of representations and similarity methods at a per-problem level. What is apparent is that there is not a policy which dominates all other methods. Every approach manages to agree with teachers the most on at least one problem. Overall, utilizing Sentence-BERT managed to have the most agreement with teachers on which student answers were similar. What is also apparent is the number of problems which Sentence-BERT performed well varied among distance measures. When using Canberra to calculate the distance between the vectors, it managed to have the highest Teacher Agreement Score with 27 out of the 67 problems. As compared to utilizing Cosine and Euclidean distance measures, which only managed to have the highest Teacher Agreement Score on 17/67 problems and 12/67 respectively. It should also be noted that the Number of Times Best Teacher Agreement Score for Problem in Table 3 will total to over 67 problems by 9. This is because there were 9 cases where two policies could be deemed acceptable for a problem; they had the same Teacher Agreement Score.

In the end, it is evident that there is not a single policy which agrees the most with teachers on which student answers are the most similar, but Sentence-BERT combined with Canberra distance is perhaps the closest of those methods explored in this work. There is a wide distribution of problems which certain method combinations outperform others, but then there are many problems in which they struggle. From this study we are able to identify those methods and problems and select when the Levenshtein ratio should be used vs Universal Sentence Encoder or the Sentence-BERT. We can use these approaches with future unseen responses (for this set of problems). By utilizing our validation results from the Teacher Agreement Scores, we can choose the best method, find the most similar current problem

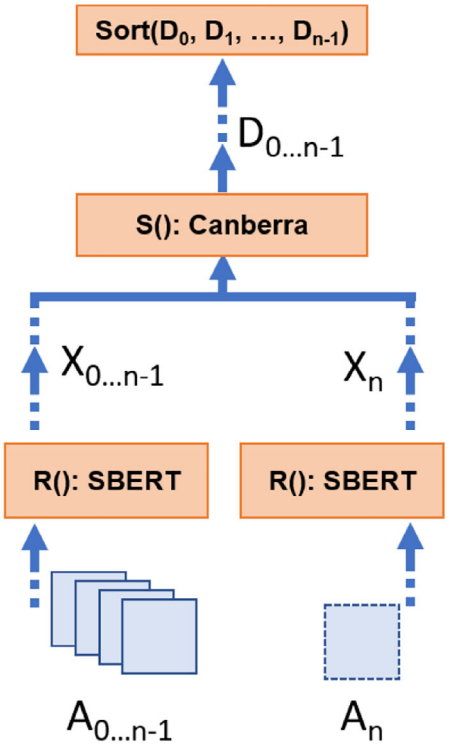

FIGURE 3 The design of the SBERT-Canberra method, that suggests scores based on similarity between the answers.

we have seen and select the teacher responses associated with that student answer as the teacher response for the new answer.

7 | STUDY 2: AUTOMATING ASSESSMENT

Following the proposed framework and results of study 1, we illustrate how this method can be utilized for the task of assessing student answers to open-ended problems. Given the previous results, we instantiate the method using a combination of Sentence-BERT and the Canberra distance measure and compare this approach to a previously-established benchmark (Erickson et al., 2020).

For this study, we use a dataset[2] composed of student answers to open-ended questions in mathematics along with the teacher-provided scores and feedback messages to these responses as used in Erickson et al. (2020). This dataset, collected from the ASSISTments (Heffernan & Heffernan, 2014), consists of 150,477 total student responses from 27,199 different students to 2076 unique open-ended questions graded by 970 different teachers. To directly compare with the methods presented in Erickson et al. (2020), we use this dataset to develop and evaluate the auto-scoring methodology. In processing the data, as was done in the previous work, we remove all responses containing only uploaded images. It is important to note that this process does leave some empty responses in the data (a total of 5,704), but are treated as any other response within this study; these empty responses will be discussed further in Study 3. Each student response is paired with a teacher-provided integer-valued score ranging from 0 to 4, with 0 being the lowest and 4 being the highest score achievable by the student for the given answer. Within the dataset, the distribution of labels is quite imbalanced, with 66.64%, 6.78%, 6.93%, 4.56% and 16.07% scored as 4, 3, 2, 1, and 0, respectively. The average number of responses to each problem in the dataset is 70.76 with an 85.63 standard deviation, and median of 47. Example student responses are included in Table 4.

Similarly as was done in the prior work, the ordinal-valued score is treated as a multiclass label, with the goal of predicting each score as a one-hot encoded vector (e.g. a score of 4 is denoted as {0,0,0,0,1}). We acknowledge that there are several limitations in framing the task in this way (as the ordering of scores is ignored in evaluating model performance), but is maintained for direct comparison to the previous work.

Using the framework described in Section 4 (illustrated in Figure 3), we convert each student answer into a 768-valued feature vector using Sentence-BERT and, for each student answer for which we want to predict a score, we compare that answer to answers within the respective training set to identify the single most-similar student response from that set. The score associated with that selected, most-similar sample is then used as the prediction for the new student answer. We also acknowledge that more sophisticated methods could be used to extend this, such as taking an average, or majority vote, but evaluate the method using this simplified approach as a means of exemplifying the framework. We do include a model that applies a regression over the top 3-most-similar answers as an additional model for comparison.

As an additional component of this model, a "fallback" condition is implemented to be able to produce scoring estimates for problems where there are no historic answers on which to compare. We acknowledge that this is an unconventional addition to such a machine learning model, as may normally be dropped from analyses (or otherwise handled in a different manner). The choice to use a fallback condition, implemented as a simpler model, comes from the intended practical use case of an automated scoring model; the model is intended to be implemented into a learning system to provide help to teachers who would normally manually score each student response. It may be confusing for the model to sometimes provide this aid and other times not (particularly within a single problem where some students receive a score and others receive nothing), and thus we chose to incorporate this fallback condition such that the model can provide a score based on simple features that have been found to correlate with scores in other problems. The use of the fallback model was needed on 65 problems, affecting 3.13% of the problems in our dataset. As this affects only the problems with the fewest sample responses, the overall impact on the analyses described in this paper is negligible. This may, however, have implications in practice if teachers utilize never-before-seen content as the fallback model is currently only applied when there are exactly 0 historic answers on

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

which to compare, but this is likely a case that could be addressed by other means (i.e. deciding not to suggest scores for never-before-seen content or defining an evidence-based threshold for including/excluding problems to support).

In this case, we train a single multinomial regression model over all known answers (across all problems), utilizing (1) the number of words in the answer and (2) the average length of each word in the answer; this model produces a probability distribution over five categorical labels (observing the 0-4 grading scale as a multinomial regression formulation). This one model is trained over all known answers and used then only in the case that no historic answers are available for the SBERT-Canberra model. This component is viewed as being part of our SBERT-Canberra approach.

7.1 | Evaluating the SBERT-Canberra scoring model

While there are few components of the described framework that "model" student work in the traditional machine learning connotation, it is worth noting that we describe this constructed SBERT-Canberra scoring method as a model given its dependence on a set of training samples (e.g. it is, in some form, modelling how teachers have previously assessed student work).

We calculate the model performance and compare our method to the previous works based on 3 performance metrics: AUC, treating the label as multinomial and calculated as described in Hand and Till (2001), Root mean squared error (RMSE) calculated over the ordinal-valued representation of the multinomial estimates and scores, and Cohen's Kappa, again using the multinomial estimates and scores. The model is trained and evaluated using a 10-fold student-level cross validation, where the model is problem-specific to compare only responses within each respective problem for similarity when generating a prediction. To evaluate these models solely based on its ability interpret the words in student responses, we make use of a 1-parameter ordinal IRT model (van Schuur, 2011), known as a Rasch model (Rasch, 1993), within the evaluation procedure, similar to that of prior work (Erickson et al., 2020); while 2- and 3-parameter IRT models observing problem discrimination and item guessing could also be leveraged to evaluate these models, the 1-parameter Rasch model used here is able to sufficiently account for student- and problem-level factors that should not be considered by the auto-scoring model and is therefore sufficient to compare model performance beyond these factors. While the output of the IRT model is not intended to be used for automated scoring itself, it does provide a structure to more fairly compare different scoring methods in their ability to understand student textual answers. The method accomplishes this by learning two parameters corresponding to one-value-per-student representing general student ability (referred to as discrimination in terms of IRT), and one-value-per-problem representing item difficulty (referred to as location in terms of IRT). As student ability and the difficulty of the item are not factors that should influence the scoring decisions of our models, IRT controls for these aspects to form a basis for comparison. The predictions of each model (i.e. the five probability predictions of our SBERT-Canberra model corresponding with each of the 5 grade scale values from 0-4) can be included into the model as additional covariates (e.g. the model will learn the IRT parameters of student ability and problem difficulty as well as beta coefficients corresponding with the five probability estimates produced by the scoring models). The performance of an IRT with these added covariates can be compared to a baseline IRT without covariates, where the magnitude of the difference describes the scoring model's ability to assess the student answer independent of the student's ability and the difficulty of the problem.

TABLE 4 Sample student responses (selected from across multiple problems for illustrative purposes) and the teacher-provided scores on a scale of 0 to 4

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

7.2 | Study 2 results

For the auto-scoring method we developed in this work, we compare the methods directly to the works from (Erickson et al., 2020) and the results are presented in the Table 5. In addition to the SBERT-Canberra method previously described, we also compare another formation of this approach. This method, referred to as "SBERT-Canberra (top 3)," uses the SBERT-Canberra method to identify the three most similar student responses to a given student answer (as opposed to the single most-similar historic answer as described for the base SBERT-Canberra model). The teacher-given scores for these top three-most-similar responses are included into a multinomial logistic regression to produce five probability estimates corresponding with each of the integer grade scale values of

Model	AUC	RMSE	Kappa	Off-by-one kappa
Current paper				
IRT* + SBERT-Canberra	**0.851**	**0.591**	**0.469**	**0.484**
IRT* + SBERT-Canberra (top 3)	0.850	0.583	0.472	0.480
Erickson et al. (2020)				
Baseline IRT	0.827	0.709	0.370	0.380
IRT + Number of words	0.829	0.696	0.382	0.395
IRT* + Random forest	0.850	0.615	0.430	–
IRT* + XGBoost	0.832	0.679	0.390	–
IRT* + LSTM	0.841	0.637	0.415	–

TABLE 5 IRT model performance for the auto-scoring method compared to previously-developed models. *IRT models also included the number of words as a predictor

zero through four. The inclusion of this method for comparison allows us to further understand how incorporating less-similar responses, as determined by our approach, impacts model performance.

This results suggest that the proposed method of SBERT-Canberra to predict a score for the student answer, outperforms the previously developed methods in (Erickson et al., 2020) across all three evaluation metrics. The Kappa value suggests that teachers are likely to agree with the score prediction from our method 47% of the time accounting for random chance, and from the RMSE, the score predictions from our model are likely to be wrong by just over half a grade point on average. While the difference in the AUC score between the previous best method and the base SBERT-Canberra method is notably small, the larger differences in other measures indicate that this approach makes improvements upon the prior methods. In observing the top-3 formulation of SBERT-Canberra, we observe inferior performance in comparison to the base formulation, but improvement over previous methods in regard to RMSE and Kappa; AUC of this top-3 formulation is found to be slightly less than the previous-best model from prior work. These findings suggest that the use of the overall most-similar answer leads to better model performance, further suggesting that our method of measuring similarity is able to rank answers in a reasonable manner.

In observing the Kappa values, it is important to acknowledge that even the highest value of 0.476 is lower than may be desired for a method intended to be used by teachers in practical settings. To help observe whether this value is an artefact of the strict grading scale observed in the study, we also report on an "off-by-one" Kappa (which treats predictions as agreeing with the label if the absolute difference between them is equal to or less than 1). Given the small difference observed, it is suggested that the model is making larger misclassifications. This is particularly interesting given the comparatively high AUC, suggesting that the model is able to distinguish between classes moderately well. This discrepancy suggests that there may be heterogeneity in the optimal rounding threshold for each score (i.e. the method for moving from a 5-valued prediction to an ordinal-scale value is seemingly sub-optimal); in terms of AUC, this may be represented by multiple intersecting curves for each class, where such crosses indicate differences in optimal classification thresholds (Ben-David, 2008). In recognizing that much of the misclassification is likely not due to off-by-one predictions, it is even more important to examine where error occurs and what characteristics of the data likely contribute to larger error; this is the purpose of Study 3 in this paper.

The relatively low Kappa may also be attributed to the subjectivity and inconsistency of teacher scoring. Gurung et al. (2022) conducted a study with teachers using ASSISTments to examine their intra-rater agreement (i.e. agreement with themselves) at different time points. In that work, teachers were asked to re-grade student work 1-2 months after initially grading the student work. It was found that teachers' agreement with themselves ranged from as low as Kappa = 0.2293 to as high as Kappa = 0.7368, suggesting surprisingly low internal consistency among some teachers. This level of variance makes it harder for an auto-scoring method to not only learn effective patterns in the data from the outcome labels, but also introduces implicit limits on how well a model can perform (i.e. if different scores are being given to semantically similar answers).

It is also just as important to emphasize that the use of the IRT model is meant purely for comparative evaluation and likely inflates these performance metrics compared to what may be expected in a real-world setting. The controlling for student ability and problem difficulty lead to higher performing models (and allow us to compare the scoring models based on their ability to assess student text while controlling for these factors), but would likely bias estimates in favour of historically high-performing students if used in practice. The SBERT-Canberra model without IRT, then, is what would be used in practice, and is observed to have an AUC of 0.70 without the IRT model (RMSE = 1.26 and a surprisingly higher Kappa of 0.53).

7.3 | Beyond correctness feedback

As has been discussed in earlier sections, it is the intention of this work to lead to better teacher supports for providing more detailed feedback to students beyond just simple correctness. Just as the methods of Study 1 contributed to the development and application of the SBERT-Canberra scoring model, the same method could be used to recommend feedback based on teacher-written feedback given to similar historic answers. In this way, the exact same SBERT-Canberra model, as its sole purpose is to rank historic responses by

similarity, can be used to select likely-appropriate messages that teachers can give in response to student answers. Following a paradigm of suggesting three possible feedback messages (i.e. as is done in Google's SmartReply; Kannan et al., 2016), the SBERT-Canberra (top-3) method explored in the previous section becomes a candidate model for this task; rather than aggregate or ensemble the top-3 responses, the model simply suggests the teacher-provided feedback previously written for these responses (continuing down the rank of similar responses if no feedback had been provided for any of the top-ranked answers).

While such a method produces potential estimates, this type of task is much more difficult to evaluate in an offline manner. From study 1, we can draw conclusions that we may expect teachers to agree with the recommendations about 62% of the time, but as TAS is a measure of teacher agreement of similarity, this may not translate to a teacher agreeing that the feedback for identified answers is necessarily appropriate.

To test the appropriateness of the SBERT-Canberra model as a feedback recommendation method, we conducted a pilot study of the QUICK-Comments Tool. Based on the SBERT-Canberra model, QUICK-Comments Tool suggested automated scores and feedback messages for open-ended responses in physical and virtual classroom environments.[3] For the collection of this data, 12 middle school mathematics teachers were compensated during the Spring and Fall of 2020 to assign assess and provide feedback to student open responses utilizing this tool; teachers were given complete freedom to score and provide feedback as they deemed appropriate and were encouraged to ignore suggested scores and feedback (overwriting these with their own) in cases where they felt the model was incorrect.

While the evaluation of the scoring component of QUICK-Comments Tool is discussed in greater detail in the next section, we found that teachers utilized one of the suggested feedback messages on 12.6% of the 30,371 student answers scored by teachers during the pilot study. While this percentage is well below the 62.3% suggested by the TAS score and indicative that there is still a large margin for improvement, this percent-age suggests that the recommendations were able to provide some utility to teachers for a portion of student responses. Considering that the Google SmartReply tool reported a usage rate of 10% (Kannan et al., 2016) in its pilot testing, we view this as a promising initial result, yet emphasizes a larger need to improve these methods, as discussed next.

8 | STUDY 3: ERROR ANALYSIS

The final study presented in this work explores our framework and, specifically, our SBERT-Canberra model to provide greater insights into its strengths and limitations; the goal of this final study is to identify characteristics of student work that may correlate with model error such that attributes with the strongest relationships can be addressed with focused improvements in future iterations of the method.

[3]The pilot study of QUICK-Comments Tool began just prior to the shift to remote learning adopted by most school systems in response to the COVID-19 pandemic and lasted through Spring of 2020.

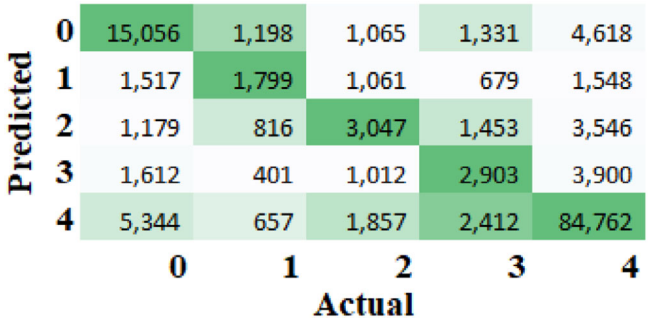

FIGURE 4 The confusion matrix of the SBERT-Canberra model after removing empty student responses.

To explore the sources of error within our model for the task of automated assessment (i.e. from observing the model and results in Section 7) we use the same dataset as used in Study 2 to conduct a set of error analyses using the SBERT-Canberra model. First, we explore the results of this model across the set of class labels using a simple confusion matrix to identify where misclassification is occurring. After this, we conduct two regression analyses using a set of answer-level features as independent variables to predict misclassification and large prediction error exhibited by the model (i.e. predictions with absolute differences greater than 0 and absolute prediction differences greater than one, respectively). The purpose of this study is to understand the potential weaknesses of the model in order to guide targeted future improvements.

Before conducting this analysis, however, we examined the predictions and labels in our data and found a counter-intuitive phenomenon regarding empty student answers present in the data. In describing the dataset in Study 2, there were a total of 5,704 empty student responses that remained in the data after following the preprocessing procedures of Erickson et al. (2020). It would normally be assumed that these empty responses would reasonably be scored as 0, given that the student did not provide an answer. In beginning our error analysis, however, it was found that the teacher-provided scores for this set of responses varied across the entire grading scale, suggesting that teachers were either scoring student work that was submitted outside the system (e.g. perhaps on paper), or teachers were scoring based on some information that could not be recorded in the system (although we find it difficult to speculate as to what the reasoning behind these scores may be); while 62.6% of these responses were scored with a 0, 29.7% were scored as 1, 4.4% were scored as 2, 1.8% were scored as 3 and 1.6% were scored as 4. For this reason, and as these clearly deviate from our intended use case in applying the model in practice (i.e. it would be infeasible for the model to anticipate and correct for this scenario when there is no student response to generate a prediction), we decided to drop these responses from the data for Study 3. The SBERT-Canberra model performance after removing these responses resulted in an AUC of 0.695, RMSE of 1.325, and a Kappa of 0.503.

TABLE 6 Features for the error analysis linear model

Title	Description	Mean
Answer length	Length of the answer	14.93
Avg. characters per word	The average number of characters per words	4.47
Numbers count	Total number of digits	2.56
Operators count	Total mathematical symbols in the response	1.89
Equation percent	Percentage of mathematical equations in answer	0.37
Presence of images	Indicator of presence of images in the answer	0.02

After removing the empty student responses, we generated a confusion matrix, shown in Figure 4, to examine where misclassification most occurs in the model predictions. From this figure, we see, unsurprisingly, that the scores are positively-skewed with a majority of responses exhibiting a maximum score of 4/4. The model appears to have the least misclassification in cases where responses are scored as either a 4 or a 0, although it is observed that the model seems biased in the direction of the majority class across all scores. We do see that the model exhibits higher off-by-one misclassification for scores of 1 and 3.

8.1 | Regression error analyses

Prior work has conducted a similar error analysis study on the same SBERT-Canberra model from Study 2 on a different dataset collected from a set of teachers who piloted an early version of the QUICK-Comments tool (Baral et al., 2021). In that study, however, it is found that the model exhibited a lower Kappa than expected despite maintaining a comparatively high AUC (AUC 0.76 and Kappa = 0.1). For this study, we expand upon the methodology utilized in that work to conduct a similar regression-based error analysis. While that work utilized a linear regression to predict prediction error from a set of answer-level features, we examine a similar approach using a logistic regression to help account for the skewed label distribution; it is our goal to examine whether the analyses lead to the same conclusions when accounting for this factor.

The regression models are based on student answer-level characteristics, comprised of a set of six answer-level features extracted from the student open response data. These features are listed in Table 6. In calculating these features, the answer is first tokenized using the Stanford NLP tokenizer (Manning et al., 2014), dividing each textual answer into smaller tokens. For example, if the response to a particular problem is "I got 2/9 by dividing by 4", a simple tokenizer splits this response text by spaces which would give the list of tokens as: ("I", "got", "2/9", "by", "dividing", "by", "4"). Then from the tokenized data, we separate the tokens consisting of either digits or mathematical symbols. The number of such tokens is divided by the total number of tokens to calculate the equation percentage.[4] The average equation percentage calculated by the procedure mentioned above is 27% across the entire dataset. For calculating the length of the answer text, we count the total words in the text simply by splitting them by space. The average length of answers across the dataset is 10.39. Similarly, within each response, the number of numeric digits (i.e. Numbers count) and number of operator characters (i.e. Operators count) are counted independent of the tokens.

ASSISTments, as a learning system, allows students to upload images as part of the response to open-ended questions; this is most commonly a picture taken of work done on paper. The response text in such cases includes the URL of the uploaded image to the system. About 15% of the total responses in the dataset contains images. While the earlier preprocessing steps removed student responses that contained only images, there are still many examples where students included images alongside other textual language. Since these scoring models are not yet designed to support images, we hypothesize that the images' presence contributes significantly to the modelling error.

We examine two logistic regression models that use the same set of features, but predicting off-by-one error and prediction error greater than one, respectively. To calculate these, we first divide the dataset into three categories consisting of samples that were correctly predicted (i.e. the difference of predicted and actual score is 0), samples that were off by one (i.e. the absolute value of the difference of predicted and actual score is 1), and all remaining samples (i.e. where the absolute value of the difference of the predicted and actual score is strictly greater than 1). We fit the first regression model to observing any degree of misclassification as the dependent variable (i.e. absolute error > 0 as the positive class and correct classifications as the negative class). This allows us to examine which response-level features may help explain any degree of misclassification exhibited by the model.

For the second regression, we dropped all of the samples that were correctly predicted and then observed the third category (where the absolute difference is strictly greater than 1) as the dependent variable; this left 36,206 responses to conduct our second regression analysis. Again conducted as a binary prediction task, this regression can be used to identify features that help distinguish between high degrees of error (i.e. absolute error > 1 as the positive class) and low degrees of error (i.e. absolute error = 1 as the negative class). As the largest possible absolute error exhibited by the model is dependent on the actual label (i.e. a true score of 2 can only exhibit maximum differences of 2 while a true score of 4 can exhibit a maximum difference of 4), we do not continue to distinguish larger error differences in additional regressions (i.e. to observe error > 2 as a dependent measure).

For both of these regressions, we report both the unstandardized and standardized beta coefficients to examine the impact of each feature on each of the observed outcomes.

8.2 | Study 3 results

The results of the error analysis of the SBERT-Canberra method are presented in Table 7. It is found that each model explains

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

TABLE 7 The resulting model coefficients for the logistic regression model of scoring error

	Error > 0			Error > 1		
	B	β	SE	B	β	SE
Intercept	−1.606***	−1.085***	0.015	0.302***	0.561***	0.026
Answer length	0.011***	0.316***	<0.001	−0.008***	−0.238***	<0.001
Avg. word length	0.016***	0.065***	0.002	0.010**	0.041**	0.003
Numbers count	<0.001	<0.001	<0.001	−0.017***	−1.125***	0.003
Operators count	−0.009***	−0.064***	0.002	0.017***	0.120***	0.004
Equation percent	0.555***	0.175***	0.021	0.808***	0.255***	0.041
Presence of images	3.445***	0.545***	0.051	1.892***	0.299***	0.064

Note: *p < 0.05, B and β denote unstandardized and standardized coefficients, respectively.
**p < 0.01; ***p < 0.001.

approximately 6% of the outcome variance as measured by a Nagelkerke pseudo-r-squared estimate (r-squared = 0.0548 for the first regression and 0.0549 for the second); this suggests that there is a large degree of variance left unexplained by our error analyses that may be attributed to other factors such as data scale per problem (as explored by Erickson et al., 2020), teacher scoring variance (as found by Gurung et al., 2022), or problem-level factors (as found by Baral et al., 2021). Despite this, however, the statistical significance and standardized beta coefficients can still identify factors that impact model error to help guide future improvements to the model.

The results of the first regression model predicting any degree of misclassification are reported on the left in Table 7. It is found that nearly all answer-level features were found to be statistically significant predictors of model error; in verifying these results, it was found that all included covariates exhibited inter-correlations less than 0.3 (suggesting a moderately low impact of multicollinearity potentially skewing the interpretation of these results). As this is a logistic regression, the coefficients are reported in log-odds units, where higher values indicate higher likelihood of a sample being included within the positive class (i.e. contributing to error) and negative values indicate higher likelihood of a sample being in the negative class (i.e. contributing to being correctly classified). While several of the unstandardized coefficients are found to be close to 0, the standardized coefficients reveal that the scale of these features changes the interpretation of their impact. In regard to this first regression, answer length, equation percent, and the presence of images in the student responses emerge as exhibiting the highest correlation with model misclassification.

The results of the second regression model predicting larger degrees of error (error > 1 compared to error = 1), are reported on the right in Table 7. In this case, all of the answer-level features emerge as statistically significant, with all but the average word length exhibiting comparably stronger relationships with the degree of error. Similar to the first analysis, the presence of equations and images emerge as contributing to larger degrees of error; the count of operators also contributes positively to higher error as well, to a lesser degree. It is found, however, that the answer length and count of numbers present in student answers contribute negatively to higher degrees of error; this suggests that these are more attributable to contributing to off-by-one error per the negative class observed in this analysis. The count of numbers exhibits an especially strong relationship with off-by-one error from this analysis.

Across both regressions, responses containing equations and images are found to have strong relationships with model misclassification, and particularly, larger degrees of error. This set of analyses aligns with the findings of prior work that examined the SBERT-Canberra model error in a slightly different context (Baral et al., 2021). Collectively, the error analyses conducted on this model suggest that future developments should target the representation of numerical values, mathematical expressions, and equations as a means of reducing modelling error. Similarly, though likely more difficult, incorporating image representations into the model may additionally help improve model performance and reduce large degrees of error; prior research has focused on building vector representations that combine language and images into the same embedding space (Harwath & Glass, 2015) and may be a direction to explore in future research. Finally, better accounting for the length of responses can help reduce off-by-one error as revealed by the second regression model. As this suggests that longer responses are related to off-by-one error (due to the reversed directionality of coefficients across the first and second regressions), it may be the case that some of these responses contain inherent distractor words that may lead to this model misclassification (Filighera et al., 2020).

9 | DISCUSSION

Considering the results of the three studies reported in this work, there are several notable characteristics of our approach that emerge. In regard to the evaluation of recommendation policies in Study 1, for example, the lack of a dominant method suggests that teachers' definition of similarity is more complex and, in also observing the error analysis of Study 3, possibly contextual. Particularly in the domain of mathematics, it is reasonable to assume that similarity will depend largely on the problem that is being observed; it is likely that the presence of certain numbers, expressions, or equations in student answers may contribute largely to whether or not a teacher would identify two answers as belonging to the same conceptual category. Conversely,

however, problems that address more abstract mathematics concepts without the use of such terms and expressions may exhibit different bases on which a teacher defines similarity. In other words, from our analyses (particularly considering the performance of Levenshtein Ratio in Table 2), we build the hypothesis that teachers consider semantic, syntactic, and mathematical attributes when grouping student answers and that the impact of these attributes may change across problems.

Although it is true that the results across analyses are relatively positive, many of these results suggest that our explored methods of representation and similarity measurement only partially align with how teachers compare student work. As identified in Study 3, the representation method appears to be targeted as one contributing factors to model error, given known difficulties of such methods and leading to more recent developments in mathematics contexts (Shen et al., 2021).

10 | LIMITATIONS AND FUTURE WORK

In regard to our approach as well as in light of our findings, there are several limitations and opportunities for future directions. In regard to the overall framework, the set of representation methods and similarity measures represent a first step toward developing more sophisticated approaches. With the re-framing of the underlying problem to be that of identifying group membership, the data utilized in the evaluation of recommendation policies could be used to train machine learning models in a more traditional manner to better learn how teachers identify similarity. Although the SBERT-Canberra approach emerged as the highest performing set of methods explored, no "training" was conducted to improve the method, which could be further explored in future works.

While the SBERT-Canberra model outperformed the prior benchmarks in assessing student open responses, the difference in performance is comparatively small. The manner in which the method makes its prediction can be considered a greedy approach in that only the closest historic answer is used to predict the score. While the inclusion of the top-3 similar answers did not lead to notable improvements, there may be better ways to ensemble similar responses beyond the single-most similar response to generate estimates. Similarly, the use of the word count model as a fallback may further be improved; while it was the case that there were arguably few instances of problems not having enough data within the cross validation, improving this fallback method may help to improve the model when applied in practical settings where the "cold start" problem is more prevalent; as the method currently relies heavily on having a sufficiently-sized pool of human-scored historic answers, future research can focus on utilizing unlabeled student answers or exploring other unsupervised methods that may additionally support these methods in cases where labelled data is scarce.

The error analysis of the SBERT-Canberra model revealed several areas where this approach, as well as others, may focus in future works. Most notably, as highlighted, the use of mathematical expressions and terms were found to be correlated with higher error; improving the representation of such elements can certainly be addressed in future work. A limitation of this, however, is that both models left variance unexplained in the outcome. We chose to look at these factors based on prior work (Baral et al., 2021), but there may be other large factors that can explain more of the error that we are seeing. Subsequent works could conduct more thorough surveys of both answer-level and higher-level factors. Future works can also explore additional model structures and language features that may lead to improvements to performance. The analyses presented in this work, however, can act as a baseline to further evaluate if future iterations of our approach truly improve upon these identified areas.

It is also the case that prior work conducting a similar error analysis (i.e. Baral et al., 2021), found that the SBERT-Canberra model exhibited conflicting performance measures when applied in a pilot study. While exhibiting high AUC metrics, Kappa was considerably low at a value of 0.1. It is unclear what the differences are between the dataset used in this work and that collected for the previous study to exhibit such a discrepancy. Future works could examine population-based and contextual differences between the two sets of studies to better understand why the differences in model performance were observed and how they might be mitigated to improve the application of these methods. Prior works in areas of machine learning have identified distributional differences between contexts to contribute largely to model performance disparities (e.g. Ocumpaugh et al., 2014; Sagawa et al., 2019). Conversely, other works have described several limitations of using AUC and kappa to measure model performance for ordinal prediction tasks, perhaps also emphasizing the need to improve how the model rectifies probabilistic estimates into ordinal predictions (e.g. by optimizing its rounding thresholds).

Extending on this, we have not deeply explored the different categories produced by teachers used to construct the TAS measure in Study 1. The clusters of student responses identified by teachers provides the data necessary to both better understand how teachers approach tasks pertaining to assessment and feedback, but also provide opportunities to explore methods of learning better similarity methods. The apparent differences in how teachers approached the task can create greater insights into the ways in which teacher assessment varies and can be examined in future work.

It is also the case that in using a pre-trained Sentence-BERT model performed reasonably well in our studies, future work could observe whether fine-tuning this model leads to improvements. Other works have started to explore the fine-tuning of BERT methods to mathematics data (Shen et al., 2021), but it is uncertain how such methods scale and generalize due to the challenges identified in the introduction; as the use of numbers and mathematical terms form an infinite set, the question is raised as to whether the set of such terms that appear in student responses forms a sufficiently-bounded set for such fine-tuning to learn meaningful representations. This question, as well as how the scale and variation of data (particularly mixing language and mathematics terms) may impact the generalized use of these language models.

11 | CONCLUSION

In considering the three studies presented in this work, the relatively positive results act as a proof-of-concept for the proposed framework which exhibited promise for application in real-world contexts. As mentioned in presenting Study 2, a teacher support tool has already undergone development and initial pilot testing utilizing these methods; while the empirical results of initial studies are still ongoing at the time of writing this paper and beyond the scope of the goals of this work, the deployment of these methods in any capacity provide suggestive evidence of their utility.

The framework itself represents an intentionally-simple structure meant to help conceptualize student modelling from an unconventional perspective. The methods used to instantiate the proposed framework are by no means novel, and neither is the concept of utilizing similarity to make predictions (e.g. k-nearest neighbour methods are based on this precise principle), but this work attempts to characterize this approach in an abstract manner to help focus research in areas of representation and the comparison of samples within a representation space. It is the presumption that such a perspective may provide utility in other educational spaces where a solution space is both sparse and vast as in the case of open response feedback.

ACKNOWLEDGEMENTS

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

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1111/jcal.12793.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Sami Baral https://orcid.org/0000-0002-6185-5841

REFERENCES

Abou-Assaleh, T., Cercone, N., Keselj, V., & Sweidan, R. (2004). N-gram-based detection of new malicious code. In proceedings of the 28th annual international computer software and applications conference, 2004. COMPSAC 2004, volume 2, p. 41-42.

Attali, Y., & Burstein, J. (2006). Automated essay scoring with e-rater V.2. *The Journal of Technology, Learning and Assessment*, *4*(3), 2-29.

Baral, S., Botelho, A., Erickson, J., Benachamardi, P., & Heffernan, N. (2021). Improving automated scoring of student open responses in mathematics. In proceedings of the fourteenth international conference on educational data mining, Paris, France.

Basu, S., Jacobs, C., & Vanderwende, L. (2013). Powergrading: A clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics*, *1*, 391–402.

Ben-David, A. (2008). About the relationship between ROC curves and Cohen's kappa. *Engineering Applications of Artificial Intelligence*, *21*(6), 874–882.

Brooks, M., Basu, S., Jacobs, C., & Vanderwende, L. (2014). Divide and correct: using clusters to grade short answers at scale. In proceedings of the first ACM conference on learning@ scale conference, p. 89–98.

Buckles, S., & Siegfried, J. J. (2006). Using multiple-choice questions to evaluate in-depth learning of economics. *The Journal of Economic Education*, *37*(1), 48–57.

Burrows, S., Gurevych, I., & Stein, B. (2015). The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, *25*(1), 60–117.

Cavnar, W. B., & Trenkle, J. M. (1994). N-gram-based text categorization. In proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval, volume 161175.

Cer, D., Yang, Y., Yi Kong, S., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., & Tar, C. (2018). Universal sentence encoder. arXiv preprint arXiv:1803.11175.

Chen, T., & Guestrin, C. (2016). Xgboost: a scalable tree boosting system. In proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, p. 785–794.

Chi, M. T., Leeuw, N. D., Chiu, M.-H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science*, *18*(3), 439–477.

Corbett, A. T., Koedinger, K. R., & Anderson, J. R. (1997). Intelligent tutoring systems.

Erickson, J. A., Botelho, A. F., McAteer, S., Varatharaj, A., & Heffernan, N. T. (2020). The automated grading of student open responses in mathematics. In proceedings of the tenth international conference on learning analytics & amp; knowledge, p. 615–624.

Filighera, A., Steuer, T., & Rensing, C. (2020). Fooling automatic short answer grading systems. In international conference on artificial intelligence in education, p. 177–190.

Foltz, P. W., Laham, D., & Landauer, T. K. (1999). Automated essay scoring: applications to educational technology. In edmedia + innovate learning, p. 939–944.

Goularte, F. B., Nassar, S. M., Fileto, R., & Saggion, H. (2019). A text summarization method based on fuzzy rules and applicable to automated assessment. *Expert Systems with Applications*, *115*, 264–275.

Graesser, A. C., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Group, T. R. G. T. R., & Person, N. (2000). Using latent semantic analysis to evaluate the contributions of students in AutoTutor. *Interactive Learning Environments*, *8*(2), 129–147.

Grenander, M., Belfer, R., Kochmar, E., Serban, I. V., St-Hilaire, F., & Cheung, J. C. (2021). Deep discourse analysis for generating personalized feedback in intelligent tutor systems. In the 11th symposium on educational advances in artificial intelligence.

Grossman, J., Lin, Z., Sheng, H., Wei, J. T.-Z., Williams, J. J., & Goel, S. (2019). Math-bot: transforming online resources for learning math into conversational interactions. AAAI 2019 story-enabled intelligence.

Gurung, A., Botelho, A. F., Thompson, R., Sales, S., & Heffernan, N. T. (2022). Considerate, unfair, or just fatigued? Examining factors that impact teacher grading. Proceedings of the 30th international conference on computers in education.

Hand, D. J., & Till, R. J. (2001). A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning*, *45*(2), 171–186.

Harwath, D., & Glass, J. (2015). Deep multimodal semantic embeddings for speech and images. In 2015 IEEE workshop on automatic speech recognition and understanding (ASRU), p. 237–244.

Heffernan, N. T., & Heffernan, C. L. (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), 470–497.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Jurman, G., Riccadonna, S., Visintainer, R., & Furlanello, C. (2009). Canberra distance on ranked lists. In proceedings of advances in ranking NIPS 09 workshop, p. 22–27.

Kannan, A., Kurach, K., Ravi, S., Kaufmann, T., Tomkins, A., Miklos, B., Corrado, G., Lukacs, L., Ganea, M., & Young, P. (2016). Smart reply: automated response suggestion for email. In proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, p. 955–964.

Kehrer, P., Kelly, K., & Heffernan, N. (2013). Does immediate feedback while doing homework improve learning? In the twenty-sixth international FLAIRS conference.

Kramarski, B., & Zeichner, O. (2001). Using technology to enhance mathematical reasoning: Effects of feedback and self-regulation learning. *Educational Media International*, *38*(2–3), 77–82.

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

Lan, A. S., Vats, D., Waters, A. E., & Baraniuk, R. G. (2015). Mathematical language processing: automatic grading and feedback for open response mathematical questions. In proceedings of the second (2015) ACM conference on learning@ scale, p. 167–176.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In international conference on machine learning, p. 1188–1196.

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

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014). The stanford CoreNLP natural language processing toolkit. In proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, p. 55–60.

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

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv: 1301.3781.

Ocumpaugh, J., Baker, R., Gowda, S., Heffernan, N., & Heffernan, C. (2014). Population validity for educational data mining models: A case study in affect detection. *British Journal of Educational Technology*, *45*(3), 487–501.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: global vectors for word representation. In proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), p. 1532–1543.

Ramos, J. (2003). Using tf-idf to determine word relevance in document queries. In proceedings of the first instructional conference on machine learning, volume 242, p. 133–142.

Rasch, G. (1993). *Probabilistic models for some intelligence and attainment tests*. ERIC.

Reimers, N., & Gurevych, I. (2019). Sentence-bert: sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

Ringenberg, M. A., & VanLehn, K. (2006). Scaffolding problem solving with annotated, worked-out examples to promote deep learning. In international conference on intelligent tutoring systems, p. 625–634.

Riordan, B., Horbach, A., Cahill, A., Zesch, T., & Lee, C. (2017). Investigating neural architectures for short answer scoring. In proceedings of the 12th workshop on innovative use of NLP for building educational applications, p. 159–168.

Roll, I., Aleven, V., McLaren, B. M., & Koedinger, K. R. (2011). Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learning and Instruction*, *21*(2), 267–280.

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

Sagawa, S., Koh, P. W., Hashimoto, T. B., & Liang, P. (2019). Distributionally robust neural networks for group shifts: on the importance of regularization for worst-case generalization. arXiv preprint arXiv: 1911.08731.

Selent, D., & Heffernan, N. (2014). Reducing student hint use by creating buggy messages from machine learned incorrect processes. In international conference on intelligent tutoring systems, p. 674–675.

Shen, J. T., Yamashita, M., Prihar, E., Heffernan, N., Wu, X., Graff, B., & Lee, D. (2021). MathBERT: a pre-trained language model for general NLP tasks in mathematics education. arXiv preprint arXiv:2106.07340.

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

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, *2009*, 1–19.

Sultan, M. A., Salazar, C., & Sumner, T. (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, p. 1070–1075.

van Schuur, W. (2011). *Ordinal item response theory: Mokken scale analysis* (Vol. 169). SAGE Publications.

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

Zhao, S., Zhang, Y., Xiong, X., Botelho, A., & Heffernan, N. (2017). A memory-augmented neural model for automated grading. In proceedings of the fourth (2017) ACM conference on learning@ scale, p. 189–192.

How to cite this article: Botelho, A., Baral, S., Erickson, J. A., Benachamardi, P., & Heffernan, N. T. (2023). Leveraging natural language processing to support automated assessment and feedback for student open responses in mathematics. *Journal of Computer Assisted Learning*, *39*(3), 823–840. https://doi.org/10.1111/jcal.12793