International Journal of Artificial Intelligence in Education
https://doi.org/10.1007/s40593-024-00418-w

ARTICLE

Automated Scoring of Constructed Response Items in Math Assessment Using Large Language Models

Wesley Morris[1] · **Langdon Holmes**[1] · **Joon Suh Choi**[1] · **Scott Crossley**[1]

Accepted: 4 July 2024

Abstract

Recent developments in the field of artificial intelligence allow for improved performance in the automated assessment of extended response items in mathematics, potentially allowing for the scoring of these items cheaply and at scale. This study details the grand prize-winning approach to developing large language models (LLMs) to automatically score the ten items in the National Assessment of Educational Progress (NAEP) Math Scoring Challenge. The approach uses extensive preprocessing that balanced the class labels for each item. This was done by identifying and filtering over-represented classes using a classifier trained on document-term matrices and data augmentation of under-represented classes using a generative pre-trained large language model (Grammarly's Coedit-XL; Raheja et al., 2023). We also use input modification schemes that were hand-crafted to each item type and included information from parts of the multi-step math problem students had to solve. Finally, we finetune several pre-trained large language models on the modified input for each individual item in the NAEP automated math scoring challenge, with DeBERTa (He et al., 2021a) showing the best performance. This approach achieved human-like agreement (less than QWK 0.05 difference from human–human agreement) on nine out of the ten items in a held-out test set.

Keywords Automated assessment · Math education · Large language models · Transformers

Introduction

Open-ended, constructed response questions are commonly used in math assessment to evaluate students' thought processes and identify misconceptions that may not be apparent using limited response questions (Kuo et al., 2016; Phelan et al., 2012). Short answer items are known to improve student math learning relative to

✉ Wesley Morris
wesley.g.morris@vanderbilt.edu

[1] Vanderbilt University, Nashville, USA

Published online: 18 July 2024

limited response items such as multiple choice (Hancock, 1995; Inoue & Buczynski, 2011; Kang et al., 2007). Constructed response items can also assign partial credit to students who understand the mathematical operation and are able to justify their thought process but fail to supply the correct final answer due to a calculation error (Slepkov & Godfrey, 2019).

Despite the pedagogical advantages of constructed response items mentioned above, they are under-used in educational contexts due to the labor required from teachers in preparing and scoring them (Hogan & Murphy, 2007; McCaffrey et al., 2022). Constructed responses are also difficult to score on learning platforms. For example, on the ASSISTments online math learning platform, less than 15% of open response items are graded and less than 4% receive feedback (Erickson et al., 2020). Constructed response math items are also rarely used in standardized testing for similar reasons (i.e., they are time-consuming and expensive to score). Thus, efforts have been made to automatically score constructed response items in math and other domains with the understanding that automated scoring would not only save resources but, more importantly, provide students with greater opportunities to engage with constructed response items (McCaffrey et al., 2022).

A recent competition hosted by the National Center for Education Statistics (NCES) attempted to address these barriers to the widespread adoption of open-ended math questions. The National Assessment of Educational Progress (NAEP) Automated Math Scoring Challenge tasked participants to build models capable of predicting human scores for constructed response answers to ten items given to students in fourth and eighth grade. The goal of the competition was to develop effective and unbiased approaches for automated scoring of open response items in mathematics (NAEP, 2021). In this paper, we describe our process of constructing the grand prize-winning models for the NAEP competition and provide suggestions for future research in the field of constructed response scoring. The goal of the paper is to explain our approach to developing models that can score constructed response items with human-like accuracy. Our hope is that the capacity to automatically score open response items at scale in mathematics will lead to wider adoption of open-ended math questions, especially in standardized assessments. This could eventually lead to the development of models that can provide feedback to students in real time outside of standardized assessments (Botelho et al., 2023) and provide information on math misconceptions that can be used by instructors in curriculum design (Nesher, 1987).

NAEP Math Challenge

NAEP is a project of the National Center for Educational Statistics (NCES) housed within the U.S. Department of Education. NAEP is tasked with measuring the achievement of American students over time in reading, writing, and mathematics. Because of the long timescale of the NAEP project, longitudinal data is available from 1971 in reading and from 1973 in math (Rampey et al., 2009). Operational control of NAEP was moved from the Education Commission of the States to the Educational Testing Service (ETS) in 1983, which implemented an item response

theory framework for norm-referenced scaled scores and introduced the concept of NAEP as 'The Nation's Report Card' (Stedman, 2008). State assessments are often mapped onto NAEP scale scores, creating a common metric which can be used to compare student achievement across states (Ji et al., 2021).

NAEP first introduced constructed response items into their assessment framework in 1992. The items comprised regular constructed response items, which required only a short answer, and extended constructed response items, where the student was asked to show their reasoning. A study by Dossey et al. (1993) found that the extended constructed response items were correlated with student proficiency and that these types of items could be used successfully at scale. However, implementing and scoring these items has required significant investment in trained raters. In response, NAEP introduced a series of data challenges in which teams compete to predict human scores of constructed response items. The first of these took place in Fall of 2021 (NAEP, 2021) and involved more than two dozen teams tasked with predicting human scores of constructed response reading comprehension problems. Five teams had prediction agreement within QWK = 0.05, considered to be within industry-accepted levels. Three grand prize winning teams achieved an average QWK score difference with human scores of less than 0.03 (Whitmer et al., 2023).

The Math Challenge, released in 2023, asked teams to predict scores for ten constructed response items in mathematics. This challenge had two goals. The first was to predict scores within QWK 0.05 of the human inter-rater reliability for each item. The second was to develop solutions that did not demonstrate bias toward or against any demographic group (race, sex, or student accommodations), defined as pairwise standardized mean differences of greater than 0.10. In addition to the prediction challenge, NAEP also introduced an innovative interpretability challenge which provided a secondary award to teams who were able to clearly interpret their models in the context of the construct they were measuring as well as perform subpopulation analyses to search for algorithmic bias and differential item performance across demographic groups (NAEP, 2023).

Automated Scoring of Student Responses

Research on automated scoring of constructed response items can help reduce the load on teachers while still providing students with valuable corrective feedback on skills. Early research focused on pattern-matching using explicitly defined language features (Sukkarieh & Blackmore, 2009; Sukkarieh et al., 2003). For instance in Sukkarieh and Blackmore's (2009) C-rater approach, each open response item was given a number of human-defined target concepts. Student responses were first automatically corrected for spelling and then parsed using algorithmic dependency parsing. Finally, the parsed response was compared against the list of target concepts and scored. Other approaches involved clustering student responses on similarity measures such as term frequency-inverse document frequency (TF-IDF, Lan et al., 2015) and latent semantic analysis similarity (LSA, Basu et al., 2013). For example, Crossley et al. (2016) developed the

Constructed Response Analysis Tool which uses LSA semantic similarity measures between student open responses and hints they receive from a science education platform. Semantic similarity measures calculated by the tool, along with other linguistic features (i.e., features related to lexical sophistication and diversity), were able to predict human scores of open responses with Cohen's Kappa of 0.404.

Neural network models such as Word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), which allow words and subwords to be represented as vectorized embeddings, can capture word-level semantics with greater fidelity than LSA and TF-IDF. Studies have used these embeddings as part of Bag of Words (BOW) models to train traditional machine learning models such as random forest and support vector machine algorithms (Erickson et al., 2020) for math constructed response. However, accuracy and generalizability remain a concern for feedback accuracy because in BOW approaches words are considered individually, outside of their syntactic context (Ludwig et al., 2021). As a result, they may not be performant enough for transfer learning and thus have lesser capacity to create generalizable models.

With the development of the transformer neural network architecture (Vaswani et al., 2017) and subsequent large language models (LLMs) such as Bidirectional Encoder Representations from Transformers (BERT), embeddings can be generated at the sentence or whole text level allowing for classification through semantic representation (Devlin et al., 2018). While Word2Vec and GloVe generate semantic embeddings for individual words, BERT takes a sequence of words or subword embeddings as its input. In the attention layers of the model, each embedding in the sequence is compared with and modified by each other embedding based on their similarity. This novel attention layer allows the meaning of words in the sequence to influence the meanings of other words. Models such as BERT produce a classification token in the final hidden layer, which contains a representation of the semantic meaning of the entire sequence, to be used for specific tasks such as classification, question answering, or to calculate semantic similarity between input sequences. After finetuning on labeled data, this token can also be used to predict human scores for constructed response items.

These types of LLMs have been used widely in recent years for scoring student responses (Lagakis & Demetriadis, 2021; Ormerod et al., 2021). For instance, Rodriguez et al. (2019) found that BERT was substantially more powerful than BOW approaches at predicting student essay quality in a publicly available Kaggle dataset, achieving a QWK score of 0.74. Similarly, Morris et al. (2023) used Longformer, a member of the BERT family capable of processing longer sequences, to predict human scores of student summary quality, explaining 79% of score variance. Transformer LLMs have also been used for automated math scoring by comparing the semantic distance of the embedding of the target response to the embeddings of all other responses and assigning the same score as the most similar response in the dataset (Baral et al., 2021). This approach achieved a degree of success, reporting Cohen's Kappa of 0.476 on a dataset of math student responses. Another member of the BERT family has been specifically trained on mathematical notation: MathBERT (Peng et al., 2021). MathBERT has been used to predict human scores on

open-ended math items, achieving strong agreement with human raters (Cohen's Kappa = 0.758) on Calculus questions (Zhang et al., 2022).

Contribution

Given the value of extended response items in math assessment, the capacity to score them automatically and at scale would have wide-reaching implications for NAEP and for the education community in general. The ability to score math items quickly and cheaply could allow them to be included in the assessment process more broadly, helping to improve feedback to students and improve learning more broadly. In this paper, we describe our training and inference pipeline for the National Assessment of Educational Progress (NAEP) 2023 Automatic Math Scoring Challenge[1] in detail and evaluate the effectiveness of our approach. We explain insights garnered from our efforts with a special attention to the problem of algorithmic bias, the use of large-language models, and the number of samples needed for developing high-quality scoring systems. Finally, we discuss some limitations of our approach, potential reasons for lower agreement on specific items, and recommend possible future avenues of research.

Methods

Training Dataset

The NAEP, 2023 Automatic Math Scoring Challenge comprises a training dataset of 251,370 responses from 10 items, five for grade 4 and five for grade 8. Each of the items required the student to provide either a short constructed response (SCR) or an extended constructed response (ECR). SCR items require students to write only a single word or digit, while ECR involve writing a sentence or more. These constructed response items (either SCR and ECR, depending on the item) are described in this paper as 'target response items' since these are the direct targets we are attempting to score. The minimum score for all target responses was 1, while the max score for target responses depended on the item and varied between 2 and 5 (see Table 1 for items).

The scores for the target responses were only one part of the total scores for each item, however, as most of the items included other responses that also contributed to the total score. In addition to the target response, all but one of the question items were multi-step items, requiring students to provide multiple responses of varying response types including short constructed responses (SCR), drop-down (DD), drag-and-drop (DaD), multiple-choice (MC), and multiple-selection (MS). In the training dataset, these responses were hand-scored separately and aggregated to form a

[1] The dataset is not publicly available, but detailed information about the dataset can be found at https://github.com/NAEP-AS-Challenge/math-prediction. Researchers can also request the dataset at https://www.researchdatagov.org/product/15704

Table 1 Item Descriptions for NAEP Math Scoring Challenge

Item	Grade	Topic	Number of Resp.	Supplemental Resp. Types	Target Resp. Type	Target Max Score	Total Max Score
ITEM 1	4	Algebraic Representations	1	None	SCR	2	2
ITEM 2	4	Patterns, Relations, Functions	2	SCR	SCR	3	3
ITEM 3	8	Mathematical Reasoning	2	DaD	ECR	3	5
ITEM 4	8	Mathematical Reasoning	2	MS	SCR	3	3
ITEM 5	4	Probability	3	MC	ECR	3	4
ITEM 6	4	Mathematical Reasoning	3	DaD, MS	ECR	3	4
ITEM 7	8	Geometry	3	SCR	ECR	2	5
ITEM 8	4	Properties of Numbers	2	SCR	SCR	3	3
ITEM 9	4	Number Operations	2	DaD	ECR	3	4
ITEM 10	8	Number Operations	2	DaD	SCR	3	3

SCR = Short Constructed Response, ECR = Extended Constructed Response, DaD = Drag and Drop, MS = Multiple Selection, MC = Multiple Choice

composite total score for each item. We use the term 'supplementary response item' to describe these additional responses since, although they are included in the dataset and they contribute to the students' total scores for each item, we are not directly attempting to predict scores for these items. Table 1 displays descriptions of the ten items in the training dataset.

In addition to the training dataset which included both labels and responses, NAEP also released a test dataset in which the student data for the extended response portion were made available, but the scores were withheld. The test dataset included 27,930 responses for the 10 items in the training set. The goal of the competition was to predict the scores of the target responses in the test dataset.

Data Set Concerns

While the goal of the competition was to score only the target short or extended constructed response portion of the NAEP item, in all but one of the items, this target response item was only part of the composite score and response. The target scores, $\boldsymbol{y}$, were sometimes partially dependent on or informed by the supplemental responses. Each item type, denoted by I, had a variable number of subcomponents. As a result, each observation, indexed by j, in the dataset, denoted as $\boldsymbol{D}$, had between one and three input features. These included the target response to score (x_{ij0}) as well as up to two additional supplemental responses in the multi-step problems, indexed by k, such that the target score (y_{ij}) depends upon $\boldsymbol{x}_{ij} = \{x_{ij0}, x_{ij1}, \ldots, x_{ijk}\}$ for all k input features.

In this formulation, each observation in the dataset, $\boldsymbol{D}_{ij}$, is a combination of an extended constructed response (ECR), supplemental responses, and a target score. Thus, the human score for the target response portion of the item is partially dependent on or informed by the score for the supplemental responses. To ensure that a

model has all of the information it needs to make an accurate prediction, previous studies have used input modification in which the model is given not only the target response but also some context with which to make its prediction (Morris et al., 2023; Raman et al., 2023). This method has shown success in previous NAEP competitions (Fernandez et al., 2022).

In addition, the target scores for most NAEP items were highly imbalanced, which can impair the accuracy of models trained on that data (Abercrombie & Hovy, 2016). One solution to this problem is to downsample over-represented classes in the training data by removing those items from the dataset (Tran et al., 2023). In addition to downsampling, data augmentation can be used, in which under-represented classes are used as models to generate simulated responses in those classes. Data augmentation can be performed either through synonym replacement at the word level (Abdullah et al., 2022) or by using generative large language models to rephrase responses of the under-represented class (Yoo et al., 2021). Data augmentation and downsampling have been shown to increase the accuracy of classification models (Bayer et al., 2023; Cochran et al., 2022; Rizos et al., 2019). Our solution was to first use junk filters and data augmentation to balance the categories, then use data modification to include all relevant information in the input during preprocessing. The advantage of this approach over simple random down sampling is that the transformer model can specialize in classifying difficult-to-score observations. Finally, we used the modified and augmented input to train separate models for each item.

Data-Preprocessing

Dealing with Data Imbalances

As noted above, some items showed significant data imbalance that could affect the training process (see Fig. 1 below for the distribution of scores across different items). These items typically showed an excess of low scoring responses (1 s) and a dearth of responses with higher scores (2 s and 3 s). It is unclear why the dataset has such a large overrepresentation of low scores, and no information was provided by NAEP. One hypothesis might be that students were unused to providing constructed responses to math items and were therefore unpracticed in the task. Alternatively, many of the low scores may be the result of low-effort responses (Braun et al., 2011; Culpepper, 2017; Finn, 2015; O'Neil et al., 2005). While random downsampling is widely used in data science (Tran et al., 2023), we chose to use a more targeted downsampling method by developing filters that could minimize data imbalance by targeting and removing responses that can be accurately assigned a low score using a simple model.

Filters

We attempted two methods to balance categories by filtering out a portion of low-scoring responses and automatically assigning them the lowest score. The first

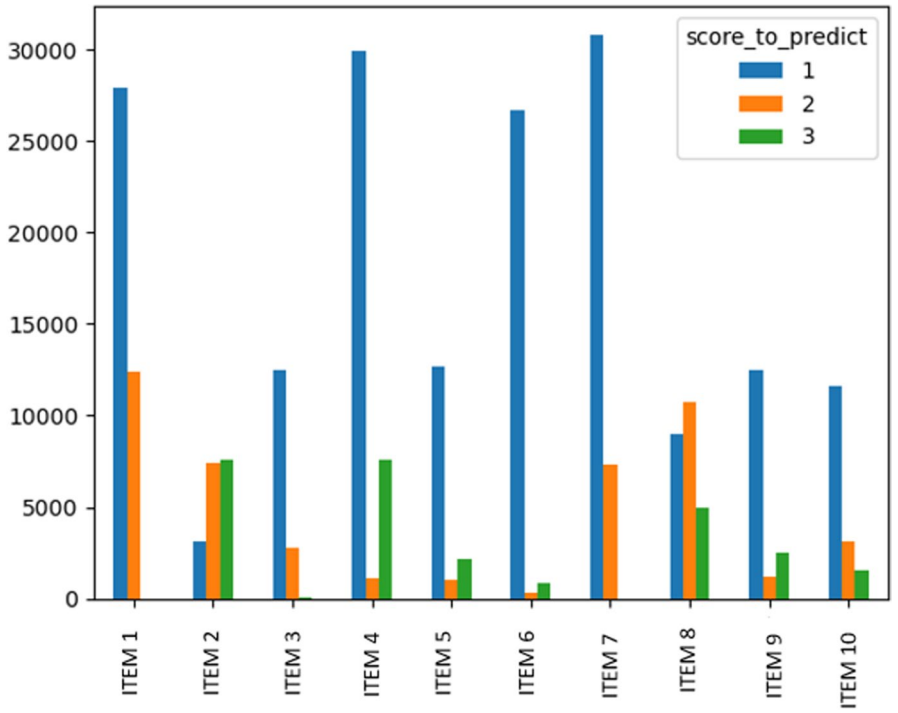

Fig. 1 Distribution of Scores across Items

Table 2 Linguistic features used as filters

Hypothesis	Features
Responses that are less likely to be written in English will have a score of 1	The probability that the response is written in English with a 0 reflecting lowest probability and a 1 reflecting highest probability
Responses with fewer words are more likely to have a score of 1	Number of words in a response
Responses with few numbers are more likely to have a score of 1	Incidents of numbers in the responses (e.g., 1, 12.2, third, three)
Responses with few symbols are more likely to have a score of 1	Incidence of symbols in the responses (e.g.,+, %, *)

method involved searching for explicitly defined language features that may be indicative of a low-scoring response and the second method involved training a logistic regression model. We first tried implementing a filter to find responses with a high probability of being low-effort answers by searching for explicitly defined language features (e.g., random string of characters, blanks, etc.). Table 2 displays the list of linguistic features used as filters and the rationale for their selection.

In addition to the approach using explicitly defined language features to filter low-scoring responses, we also trained a logistic classifier through stochastic gradient

descent (SGD) on a document-term matrix for items with data imbalances. An SGD classifier learns through gradient descent similarly to a traditional neural network model. However, instead of computing the gradient of the loss function after each batch, the SGD algorithm computes the gradient for only a single randomly chosen training example at a time. At the time it was proposed, SGD was shown to be faster and more suitable for large datasets (Zhang, 2004), and it remains a common way to train logistic regression algorithms using TF-IDF (Gaye et al., 2021; Goswami & Sabata, 2021).

Since *ITEM 2* and *ITEM 8* already had roughly balanced labels, we did not develop filters for these items. In practice, we found that the SGD classifier performed better than the linguistic features approach and the linguistic approach was dropped. SGD classifiers were trained on a document-term matrix (DTM) consisting of all tokens and bigrams that appeared in at least 2 documents, excluding a set of English stop words (Pedregosa et al., 2011). The DTM ignored case and used TF-IDF vectorization. Classifiers were trained with an alpha of 1e-4 and a logarithmic loss function. Class weights were optimized by hand to filter out as many responses that scored a 1 as possible while misclassifying roughly less than 1% of 2-scoring and 3-scoring responses. Responses of 2 and 3 were assigned a weight of 1.0, and responses of 1 were assigned a smaller weight depending on the question item. Filtered items were given the lowest score (1) and were not processed by the model during training or during prediction. In our final solution, we applied this filter to the items and assigned a score of 1 to all items that did not pass the filter.

Data Augmentation

Our next preprocessing step was to generate additional high-scoring target responses to be used as training data using the available dataset (i.e., data augmentation). Our approach was to use a pre-trained large language model to create multiple paraphrases of students' responses with higher scores (i.e., responses scored 2 or 3). After working with multiple different data augmentation libraries (Parrot, Pegasus, NLPAug, wordnet), we elected to use a large language model for data augmentation. Coedit-XL was the largest in a series of models that were developed and released by Grammarly for writing assistance applications (Raheja et al., 2023). The Coedit models were finetuned from T5, which is a pre-trained language model developed by Google (Raffel et al., 2019). According to work by Raheja et al. (2023), the Coedit models perform nearly as well as GPT-4 for paraphrasing but are much smaller. Importantly, the Coedit models are publicly available and can be run locally, a requirement for privacy-protected data such as the NAEP data.

We generated a dynamic number of augmentations per item and per score-to-predict, attempting to roughly balance the number of responses per score and per item. A maximum of 8 augmented responses were generated from each authentic student response. Paraphrases were generated using the Coedit-XL model with beam search and a 'diversity penalty' to encourage better diversity in the generated sequences. In the process of paraphrasing, the Coedit-XL model tended to correct the spelling, punctuation, and capitalization in students' responses. To better reflect the writing characteristics of authentic student submissions, we randomly injected one

spelling error into 50% of the augmented samples and fully lowercased 50% of the augmented samples (the same sample could be augmented with a spelling error and lowercased). Spelling errors were generated using NLPAug (Ma, 2019), a Python package that includes a list of common English spelling mistakes.[2]

Input Modification

Some question items contained additional supplemental response items from students that were not recorded in the target response column but were recorded elsewhere in the dataset. Although our goal was to score the target responses, not the supplemental responses, the human raters had access to the supplemental responses when they were scoring the target responses, and the scores for the target responses were sometimes dependent on the context of the supplemental responses. Thus, it was necessary to include information about the supplemental responses in the input. Table 1 shows the types of supplemental responses present in each item. In this section, we will discuss how we generated modified inputs for each item based on the question type and the format of the responses that were provided.

First, we identified items that included supplemental responses in their scoring process, as specified in the provided item description documentation. These were all items except for *ITEM 1* and *ITEM 3*. For the remaining 8 items, we extracted relevant information from other columns and constructed a modified input sequence so that the input sequence x_{ij} is composed of a concatenated string. We organized the supplemental responses into three categories: text responses (aside from the target response item), selection responses, and drag-and-drop responses.

Three items contained text responses in the supplemental response fields. These are *ITEM 2*, *ITEM 7*, and *ITEM 8*. While the supplemental response fields associated with *ITEM 7* do not directly impact the score to predict, we found that these fields were informative about the score to predict: students who were scored as responding correctly to supplemental Parts B and C were more likely to be scored as correct on target Part A. As a result, we opted to include this additional information in our modified input. We constructed the modified input by prepending each part of the student's response with a label, e.g., "Part A: $\{x_{ij0}\}$ Part B: $\{x_{ij1}\}$ Part C: $\{x_{ij2}\}$". The large language models we developed later to score the responses were trained on this modified input, rather than only on the provided target response (although the target response is included in the modified input as x_{ij0}).

ITEM 2 and *ITEM 8* included supplemental response text fields that directly impact the score to predict. A correct answer is normally a single digit number represented as a string ("23" and "95", respectively). However, many students submitted alternative responses that human raters scored as correct, such as "143–48 = 95" for *ITEM 8*. We opted to include the student's raw response in the input sequence, reasoning that our transformer-based scoring models would learn to utilize simple regex-like patterns. However, unlike a hand-crafted regex scoring system, the transformer may also learn more complex relationships between the supplemental

[2] https://github.com/makcedward/nlpaug/blob/master/nlpaug/res/word/spelling/spelling_en.txt

response field and the target response field (x_{ij0}). For example, in *ITEM 2*, the response "3" was scored as correct by human raters 52 times (and scored as incorrect 1,143 times), with the few responses that were scored as correct frequently containing some correct rule in x_{ij0}. We presumed that our large language models may acquire more human-rater-like behavior by learning to predict using both columns. For both items, we construct inputs in the form "Part A: $\{x_{ij1}\}$ Part B: $\{x_{ij0}\}$".

ITEM 4, *ITEM 5*, and *ITEM 6* contained selection-type supplemental responses (e.g., multiple choice answers) that directly impacted the score to predict. Information on whether the student responded correctly to the multiple-choice questions was prepended to the student's written response in written form (i.e., "Part A: The student definitely responded correctly Part B: $\{x_{ij0}\}$"). However, this method was made more complicated by inconsistencies in human scoring of the supplemental response items. For example, *ITEM 4* exhibits inconsistent scoring for some student responses (e.g., the response "1:2" was equally as likely to be scored as correct or incorrect by human raters). On the other hand, when a student responded with "c(1, 2, 4)", there was a 99.5% chance that a human rater would consider this response correct. Due to the large sample size, we opted for a data-driven approach to scoring supplemental response components. We constructed input sequences that reflect the consistency in human rater scoring for each response type. To return to the previous example, in which half of the human raters scored a response as correct, we used the input sequence, "Part A: The student maybe responded correctly. Part B: $\{x_{ij0}\}$." In cases where the vast majority of human raters scored the response as correct, the input sequence would be, "Part A: The student definitely responded correctly. Part B: $\{x_{ij0}\}$". We constructed responses in this manner using a Likert-style scale. The Likert-style textual representation was a function of the probability that a given response (x_{ijk}) would be scored as correct:

$$L(x_{ijk}) = \begin{cases} \textit{definitely incorrect}, & p(x_{ijk}) \leq .02 \\ \textit{probably incorrect}, & .02 < p(x_{ijk}) \leq .30 \\ \textit{maybe correct}, & .30 < p(x_{ijk}) \leq .70 \\ \textit{probably correct}, & .70 < p(x_{ijk}) \leq .98 \\ \textit{definitely correct}, & .98 < p(x_{ijk}) \leq 1.0 \end{cases}$$

These thresholds (0.02, 0.30, 0.70, 0.98, 1.0) were chosen to reflect our own intuition about scoring confidence and were not optimized. We utilize this same input construction pattern for all supplemental response question components.

Some supplemental response components were stored across multiple columns. *ITEM 10* and *ITEM 9* have drag-and-drop responses that directly impact the score to predict. For these questions, we implemented a similar approach as the selection-type items. However, we first had to represent the response as a single value because the students' responses to drag-and-drop question components were spread across 6–8 "source" and "target" columns. Figure 2 presents an example of a drag-and-drop style item with four source elements and three target elements. We first converted all the relevant columns into a single string representing the student's full response, e.g., "12344321". We then tallied the number of times a human rater scored this string as correct/incorrect and designed an input sequence that reflects our confidence about

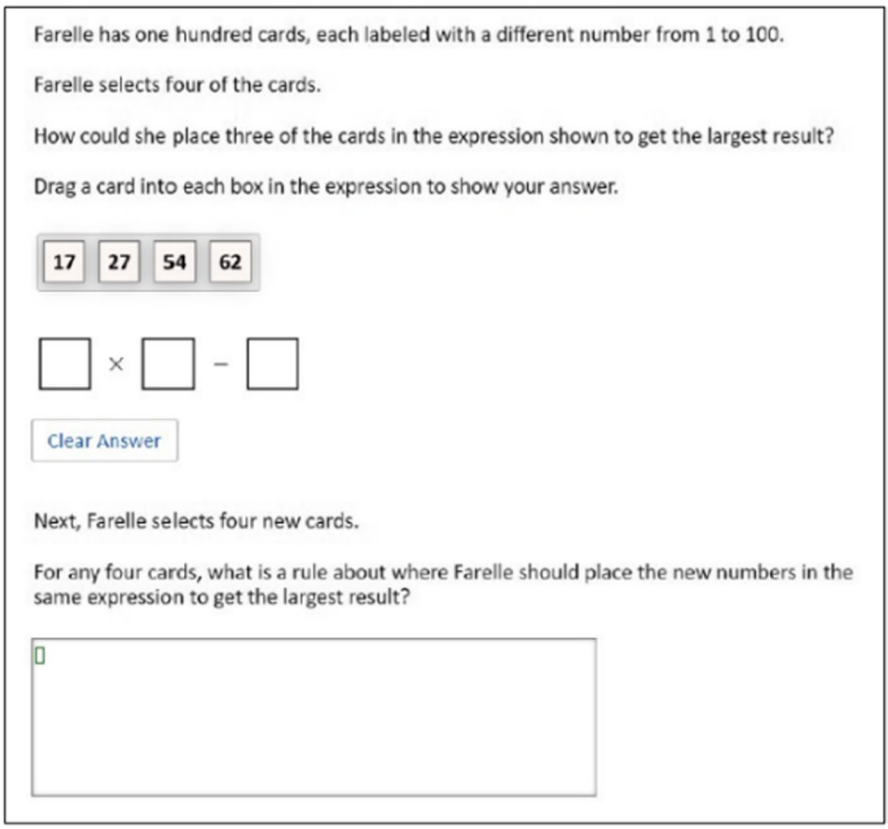

Fig. 2 Example of drag-and-drop item

the correct score for this response string using the same approach described for the selection-type items.

Training Attempts

Train/Validation Splits

After preprocessing, the full dataset was split by item ID, and each item was further divided into training and evaluation partitions with a 70/30 split with proportionate score allocation. The evaluation partition was also used as the testing set, wherever results are reported for these models. Where noted, models were trained using five-fold cross validation, which amounts to an 80/20 split for each fold. Every sample was part of exactly one model's out-of-fold evaluation partition. On the held-out testing set, the predictions of the five models were combined using mean-pooling. The combination of the cross-validated models allowed us to leverage all available data on the test set, although each model had to be validated separately on the

training data, meaning that each individual model only had access to 4/5 of the total data.

Encoder-Only Models

After experimenting with several embedding models including MathBERT (Peng et al., 2021) and MPNet (Song et al., 2020), we settled on finetuning DeBERTa (Decoding-enhanced BERT with Disentangled Attention, He et al., 2021a, 2021b) on each individual item to create baseline models. We structured training as a pseudo-regression task, in which the model was fitted with a linear layer for classification with a single output label whose logit corresponded to the score to predict. During training, loss was calculated as the mean squared error between the model predicted score and human-rater score. Because there are no fractional scores in the dataset, the predicted scores based on pseudo-regression were rounded to the nearest integer during post-processing. During evaluation, we selected the model which exhibited the highest quadradic weighted kappa between the predicted scores and the human-rater scores.

In addition to the pseudo-regression-type sequence classification task described above, we also experimented with reframing the problem as a categorical classification task. In this case, each possible score level depending on the item $y_i \in [s_{min} = \{1\}, s_{max} = \{2, 3\}]$ is assigned its own output label. One advantage of reframing the problem as a categorical classification task is that this formulation uses cross-entropy loss, a common loss function in deep learning. Unlike mean-squared error, cross-entropy loss can be weighted by per-label sample size during training, which we thought might improve the model-human kappa.

The model hyperparameters incorporate parameters of the model which are manually set rather than learned through the training process, including number of epochs trained, learning rate, weight decay, and dropout probability during training. To select the optimal hyperparameters for our model, we trained models using different settings through a grid search with Bayesian optimization. Initial results suggested that optimal hyperparameter settings were consistent across items, with learning rate being the most important parameter. Based on the results of the hyperparameter optimization, we set the learning rate to 1.5e-5. Weight decay and the number of epochs had less impact on the final model performance. We set these to 0.3 and 2. This process of hyperparameter tuning may represent an improvement on similar studies in which it was not performed and the default settings were used for hyperparameters (Baral et al., 2021). We then carried out training using DeBERTa-v3-large. The decision to use the DeBERTa line of models was largely informed by reviewing top performers in relevant Kaggle competitions, such as the Feedback Prize (Wang et al., 2022).

Model Bias

We acknowledge that bias is an important consideration, especially when dealing with black-box language models (Baffour et al., 2023). Thus, we assessed the potential for our models to exhibit bias in a post-hoc comparison. In this analysis, we used

a version of our final models that were trained with all the same data and hyperparameters as the final models, except using a hold-out validation group to ensure that the test data and the training data remained separate. This contrasts with the cross-validation approach used in the final models, which would have added a layer of complexity to the bias analysis. We first calculated and evaluated all pairwise Standardized Mean Differences (SMD), or the difference between the two means divided by the pooled standard deviation, between different demographic groups. We followed up the pairwise SMD approach by building a linear mixed effects regression model to investigate the effect of the demographic variables on model error. Prediction error here is defined as the difference between the score predicted by the language model and the score assigned by human raters. Our goal was to evaluate whether any of the demographic variables had a significant effect on prediction error. In the mixed effects model, responses were grouped by item as random effects so that we could observe the effect of the demographic variables with greater fidelity.

Testing

After finetuning the models on the training dataset, we used the finetuned models to predict the scores of the target response items of the test dataset. The scores for the test dataset were withheld from the training set, and the scores generated by our finetuned models on the test set responses were sent to NAEP to be independently evaluated. Quadratic weighted kappa between the model-assigned score and the human-assigned score was used as the performance metric, with a difference of $QWK < 0.05$ considered within an acceptable limit for human-like performance. The NAEP data science team also evaluated pairwise SMD between demographic groups to ensure no bias in the models.

Results

Filtering Low-Scoring Responses

We found that the approach of using explicit language features present in the responses to identify low-effort responses worked reasonably well at eliminating low-scoring responses but it also removed non-negligible portions of high-scoring responses resulting items being misclassified as 1 s. Figure 3 below shows the percentage of differently scored responses being removed as we applied the language probability filter with different thresholds to one of the items. Because of the tendency of language features to misclassify and remove high-scoring responses, we determined that this method was inadequate to our purposes Fig. 4.

Our other method of classifying low-scoring responses, the SGD classifier on a document-term matrix for items with data imbalances, performed better. Table 3 below shows weights assigned to 1-scoring items, the percentage of 1 s that were successfully filtered out, and the percentage of 2 s and 3 s (if any) that were lost. This method was much more precise than using explicitly defined language features,

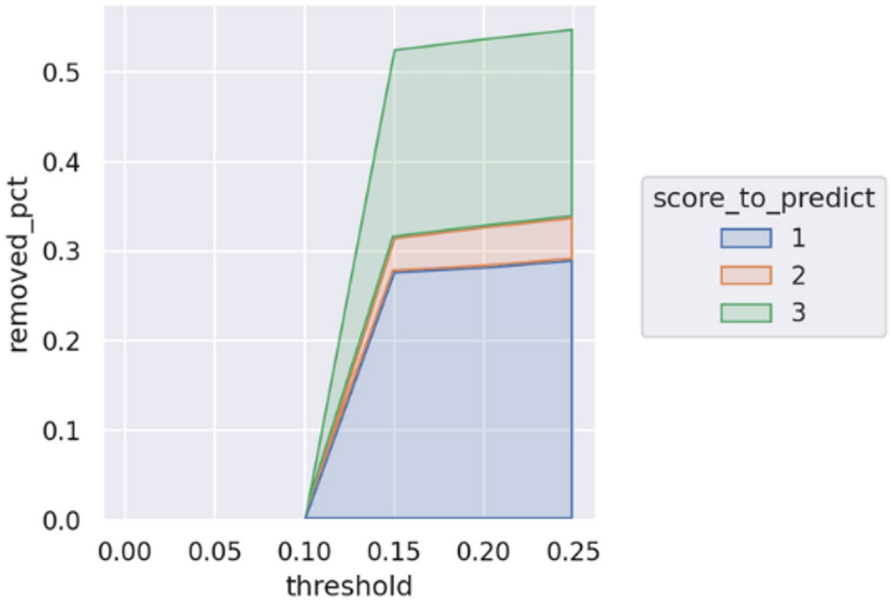

Fig. 3 Percentage of Scored Responses Removed with Different Thresholds for "Language Probability"

and as a result we chose this for our final submission. Our model automatically assigned items identified to be low-scoring scores of one, and these items were removed from the train and test datasets.

Data Augmentation

After using the SGD classifier to filter out low-scoring responses, we were still left with notably imbalanced data for several items. To further balance the classes of the training data, we used Grammarly's Coedit-XL model to augment our dataset. The exact number of artificial responses generated per actual response was dependent on the level of imbalance in the dataset. Table 4 below shows the items where this data augmentation approach was used as well as their configuration. Although the datasets remained unbalanced even after data augmentation, especially in terms of the high prevalence of low-scored responses, we determined that the model could not produce more than eight diverse, high-quality paraphrases of the same text Fig. 5.

Training

Table 5 below shows the quadradic weighted kappa scores from MPNet, MathBERT, and DeBERTa-v3-large compared to labels, as well as the human-to-human QWK scores as a benchmark. In the interests of time and compute, these preliminary models were trained using a hold-out partitioning strategy rather than cross-validation. MPNet performed nearly as well as DeBERTa-v3-large for most items, which could

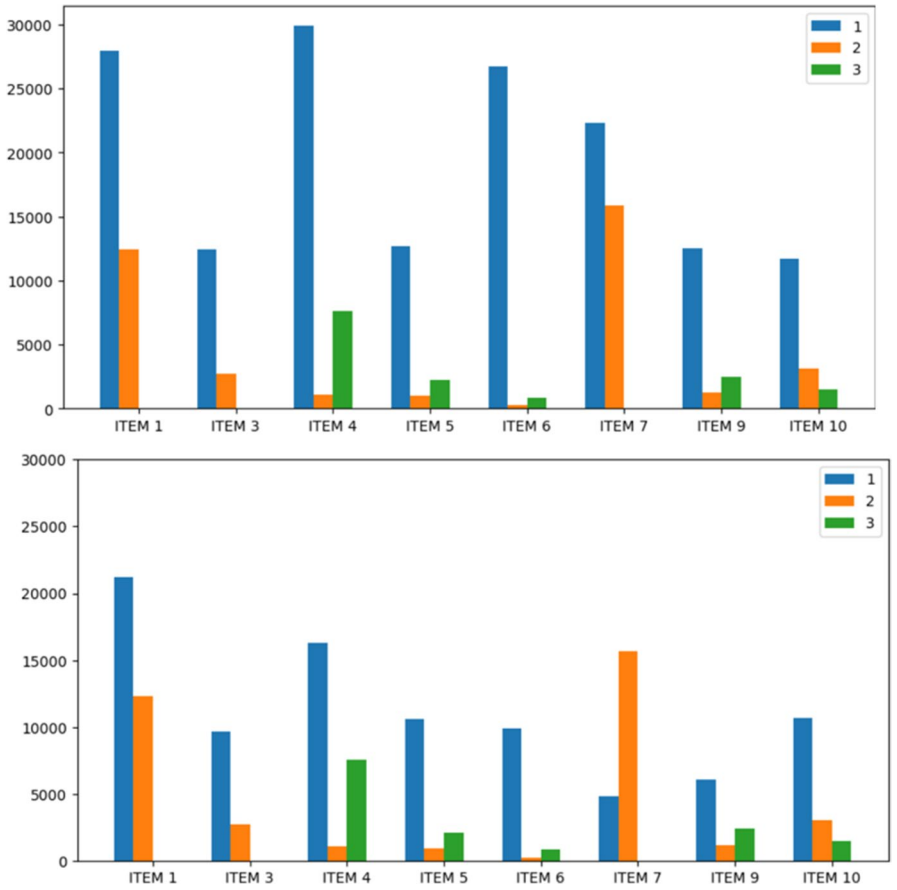

Fig. 4 Data Balance pre- and post-Filtering (top to bottom)

Table 3 Filter types, accuracy, and thresholds

Item	Weight	Filtered 1 s	Retained 2 s/3 s
ITEM 1	3.00E-02	24.13%	99.36%
ITEM 2	–	–	–
ITEM 3	3.00E-02	22.24%	99.14%
ITEM 4	2.00E-02	45.41%	99.36%
ITEM 5	2.50E-02	15.70%	98.53%
ITEM 6	5.00E-03	62.98%	98.87%
ITEM 7	5.00E-03	78.14%	98.87%
ITEM 8	–	–	–
ITEM 9	3.00E-02	51.34%	99.02%
ITEM 10	6.00E-02	8.36%	99.38%

Table 4 Number of Augmented Samples Generated from each Provided Sample, Based on the Item's Provided Score to Predict

Item	Number of Augmented Samples Generated per Response		
	Score of 1	Score of 2	Score of 3
ITEM 1	0	0	–
ITEM 2	0	0	0
ITEM 3	0	0	0
ITEM 4	0	7	0
ITEM 5	0	4	2
ITEM 6	0	8	3
ITEM 7	0	0	–
ITEM 8	0	0	8
ITEM 9	0	2	0
ITEM 10	0	0	3

make it an appealing choice in a production setting where computational costs need to be balanced with accuracy. MathBERT underperformed compared to DeBERTa-v3-large. Considering these results, we chose DeBERTa-v3-large as our base model for finetuning across all items.

Loss Function

Our initial attempts at framing the task as a category classification problem rather than a regression problem were unsuccessful. We found that mean-squared error (MSE) loss was at least as effective as cross-entropy (CE) loss for almost all items as reported in Table 6 below. This may be because MSE penalizes large discrepancies between predicted and actual scores more than smaller differences. These loss functions were compared with fivefold cross-validated models developed from DeBERTa-v3-large.

Final Model Performance on Training Set

Our best performing models were trained using fivefold cross-validation on the full training dataset using stratification to ensure equal distribution of scores across folds. Some models' training pipelines also include data preprocessing filters and data augmentation. Thus, the scoring model for each item is an ensemble of five models, each trained on 80% of the dataset and retaining 20% as a validation set. In the final model, mean pooling is used across all five models to calculate the final prediction. The training configuration for each of the models as well as their average performance on out-of-fold training data are shown in Table 7 below. The final column displays the difference between the model-to-human QWK and the human-to-human QWK. More details regarding the configurations of the filters and data augmentation used for each individual item can be found in Tables 3 and 4 respectively.

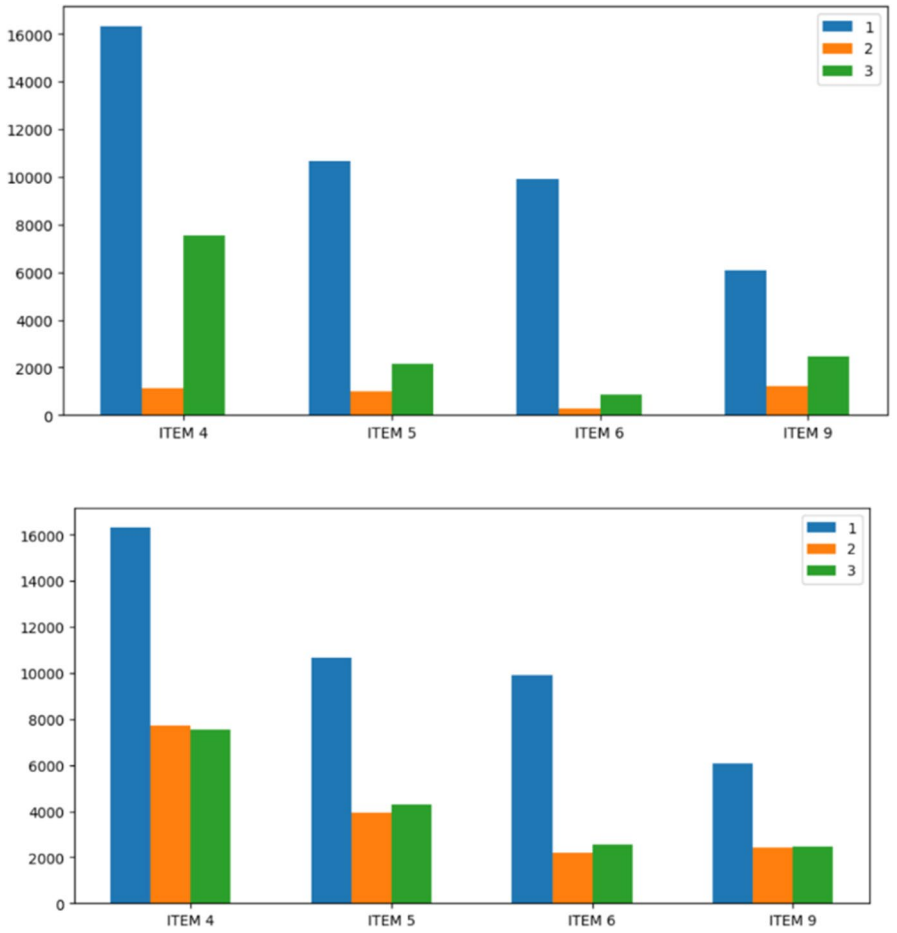

Fig. 5 Data Balance pre- and post-Augmentation (top to bottom)

Table 5 Comparisons across embedding classification models

	Human-to-human	Deberta-v3-large	MPNet	MathBERT
ITEM 1	0.966	0.941	0.935	0.926
ITEM 2	0.981	0.962	0.963	0.963
ITEM 3	0.910	0.882	0.862	0.852
ITEM 4	0.933	0.871	0.860	0.832
ITEM 5	0.948	0.906	0.897	0.877
ITEM 6	0.946	0.828	0.815	0.776
ITEM 7	0.980	0.943	0.936	0.934
ITEM 8	0.984	0.983	0.983	0.983
ITEM 9	0.992	0.960	0.942	0.938
ITEM 10	0.956	0.925	0.926	0.913
mean	**0.960**	**0.920**	**0.912**	**0.899**

Table 6 Model Performance (QWK) for CE and MSE Loss Functions

Item	Human QWK	Weighted CE QWK	MSE QWK
ITEM 1	0.966	0.938	0.939
ITEM 2	0.981	0.961	0.961
ITEM 3	0.910	0.879	0.883
ITEM 4	0.933	0.835	0.851
ITEM 5	0.948	0.780	0.903
ITEM 6	0.946	0.815	0.687
ITEM 7	0.980	0.920	0.916
ITEM 8	0.984	0.871	0.983
ITEM 9	0.992	0.931	0.930
ITEM 10	0.956	0.946	0.937
mean	**0.9596**	**0.8876**	**0.899**

Table 7 Final Models: Configurations and Performances

Item	Filters	Data Augmentation	Out-of-fold QWK	Human QWK	Model—Human Difference
ITEM 1	O	X	0.939	0.966	-0.027
ITEM 2	X	X	0.961	0.981	-0.020
ITEM 3	O	X	0.879	0.910	-0.031
ITEM 4	O	O	0.850	0.933	-0.083
ITEM 5	O	O	0.898	0.948	-0.050
ITEM 6	O	O	0.846	0.946	-0.100
ITEM 7	O	X	0.916	0.980	-0.064
ITEM 8	X	O	0.983	0.984	-0.001
ITEM 9	O	O	0.928	0.992	-0.064
ITEM 10	O	O	0.937	0.956	-0.019

Post-Hoc Analysis of Bias in Training Data

No pairwise Standardized Mean Difference value was greater than 1.6e-8, indicating no evidence of bias toward or against any specific demographic group contained in the training dataset. We further investigated the possibility of algorithmic bias by constructing a linear mixed-effects model to investigate prediction error, defined as the difference between the scoring model's predictions and the actual scores in the training set. Bias in error could indicate that the model is consistently over or under-scoring for specific demographic groups. The only effect which reached the threshold for statistical significance was for American Indian/Alaska Native ethnicity, which scored an average of 0.02 points lower than the baseline category of White/Caucasian (SE = 0.006). While statistically significant, this effect is small and may be a result of the relatively small sample size for this group ($n \approx 1{,}590$). The

Table 8 Mixed Effects Model Including all Demographic Variables, Grouped by Item

Predictor		Est	Std.Err	df	*t*	*p*
	Intercept	0.031	0.014	10	2.299	0.044
Race/ Ethnicity	African Amer, not Hispanic	-0.005	0.003	67,920	-1.854	0.064
	Hispanic of any Race	-0.002	0.003	67,920	-0.914	0.361
	Asian, not Hispanic	-0.006	0.005	67,920	-1.233	0.218
	Amer Ind/ Alaska Native	-0.020	0.006	67,920	-3.070	0.002
	Native Ha/ Pacific Island	0.001	0.011	67,920	0.058	0.954
	> 1 race, not Hispanic	-0.008	0.005	67,923	-1.688	0.091
Sex	Male	0.000	0.002	67,920	-0.093	0.926
Accom	Without Accommodation	0.003	0.005	67,920	0.675	0.500
IEP	Without IEP	0.003	0.004	67,920	0.604	0.546
LEP	Not English Learner	0.001	0.004	67,930	0.231	0.817

Table 9 Comparison of model agreement with human score to human agreement with human score QWK

Item	Model QWK	Human–Human QWK
ITEM 1	0.949	0.966
ITEM 2	0.961	0.981
ITEM 3	0.890	0.910
ITEM 4	0.889	0.933
ITEM 5	0.945	0.946
ITEM 6	0.841	0.928
ITEM 7	0.950	0.980
ITEM 8	0.986	0.984
ITEM 9	0.962	0.986
ITEM 10	0.963	0.956
Average	**0.945**	**0.965**

marginal R^2, or the proportion of variance in prediction error that can be explained by the demographic variables after controlling for the item effect, was 0.0003, which is near zero. Based on the results of the pairwise standardized mean differences and the results of the mixed effects model, we concluded that there is not sufficient evidence to suggest that the model was meaningfully biased towards or against a demographic group Table 8.

Performance on Test Set

Table 9 displays the performance of our models on the test set. Our models were more accurate in the test set (mean QWK = 0.945) than in the training dataset (mean QWK = 0.899). This result is likely because the internal evaluation models were each ensembles of five models that each used only 80% of the data and tested

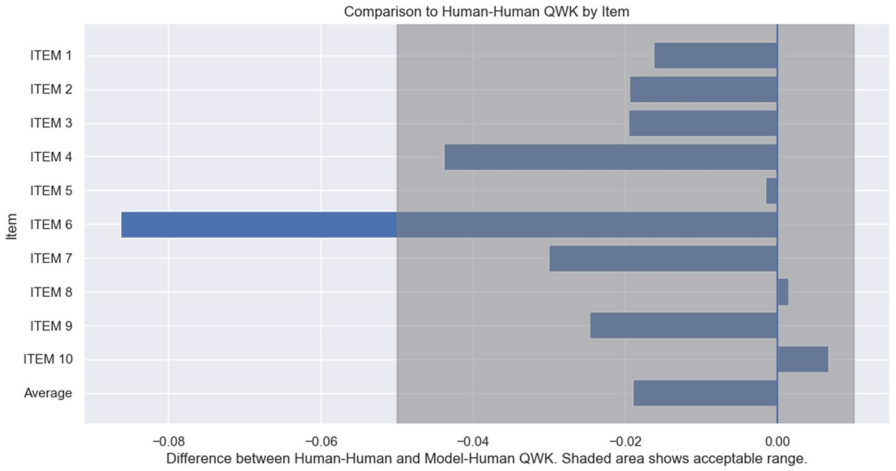

Fig. 6 Comparison between human–human agreement and model-to-human agreement by item. Shaded area represents acceptable agreement (< 0.05 QWK difference from human–human agreement.)

against the 20% of the data which were out-of-fold during fivefold cross validation. However, the final models included all five folds generated during the cross-validation approach. Thus, the full dataset was leveraged to train the final models. This improvement in performance may demonstrate the power of using cross-validation over a hold-out train-valid-test split. Additionally, the improved performance in the final model indicates that we did not overfit our model to the training data, and that it generalizes well to data outside of the training dataset.

Differences between model-human QWK and human-to-human QWK by item (the criteria on which our performance was judged) are displayed in Fig. 6. The shaded gray area represents an acceptable difference of less than 0.05. Our models were within acceptable agreement for nine out of the ten items, with only ITEM 6 having lower than human-like performance.

Discussion

Our solution to NAEP's, 2023 Automatic Math Scoring Challenge relied on data preprocessing techniques along with finetuning a transformer-based language model for each item. The solution achieves scores that were comparable in reliability to human raters for nine out of the ten items. These results point to the potential to use large language models in math assessment, including scoring extended response items in which students are asked to explain their reasoning. This type of math assessment has valuable pedagogical implications and the capacity to deploy it cheaply and at scale could have an important impact on math learning. In addition, our bias analysis showed that our solution was not meaningfully biased against any demographic group. Below, we outline ideas on how this study contributes to the field of automated constructed response item scoring in mathematics.

We have provided detailed notes on our decision-making process for constructing our automatic scoring models. We started by pre-processing the data to deal with data imbalances. We found that filters using features such as language probabilities, number of words, and incidence of numbers or symbols in the responses performed reasonably well, but removed non-negligible portions of high-scoring responses, which limited their utility. We had greater success using SGD classifiers for items with significant data imbalances. We also used data augmentation to generate additional high-scoring responses as training data, and modified inputs for certain items to include necessary information which was not initially included in the target constructed response. After experimenting with different loss functions and different pre-trained models, we finalized our model's configuration and parameters. Our final models were DeBERTa-v3-large models finetuned using fivefold cross-validation on stratified folds with varying respective configurations for each item regarding the use of filters and data augmentations.

Six out of the ten final models in the training data (trained on fivefold cross-validation) produced sufficiently accurate (within QWK 0.05 of the human inter-rater reliability) predictions for out-of-fold items. The models for *ITEM 4, ITEM 6, ITEM 7*, and *ITEM 9* failed to produce sufficiently accurate predictions on our internal evaluation set. In the test dataset, which was withheld by NAEP until the end of the competition, our models achieved sufficiently accurate predictions on nine out of the ten items, only reporting lower-than-acceptable predictions on ITEM 6. The improved performance of the models on the test set relative to the training set is probably because the predictions on the test set were from five different models, each trained on a different subset of the training data, with the predictions of the five models averaged. The results on the internal evaluation set were only the average accuracies of each individual model. As a result, the full dataset was leveraged in training the ensemble model for the test set, while each individual sub-model only had access to 80% of the training data.

The sub-optimal performance of the model for *ITEM 4* in the internal evaluation set is possibly related to missing or incomplete data. Whether or not the student selected the correct multiple-choice option is an integral part of the scoring process for this item. However, students' actual choices were available for one subset of the data while students' selection in the other subset of the data could only be partially inferred. The poor performance of the model for *ITEM 6* in both our internal evaluation set and in the test set is likely due to data imbalance. Along with *ITEM 3*, this item had the smallest proportion of higher-scoring student responses as shown in Fig. 1. Data augmentation and filters helped balance out the data and improved the model performance to some extent. Additionally, there was a high degree of variability in the human-rater scoring of the supplemental responses for this item. Specifically, the response "1:2" had roughly a 50% chance of being scored as correct or incorrect, which may indicate that the scoring for this item was less reliable than for other items, or perhaps dependent upon external information that was not included in our development process.

While the provided data was ample for most items, we believe that more accurate models could be developed with additional samples for *ITEM 6* and *ITEM 3*. These items had fewer than 1,000 samples for the higher scores, which negatively

affected the performance of the models we trained, and we also suspect that some items have more data than is necessary to develop highly effective models. We did not experiment with random downsampling strategies ourselves, but this approach could provide valuable information about the number of samples needed to develop stronger models. We did, however, apply filters that removed upwards of 50% of the 1-scoring samples from some items (meaning that these samples were not seen by our models during training), and this markedly improved model performance, suggesting that class balance and the distribution of the data are more important than the overall number of samples collected.

It bears noting, again, that no demographic information was used in our training pipeline. Pairwise Standardized Mean Differences between different demographic groups and a visual inspection of the mean prediction errors showed that there were no significant biases in our models' predictions beyond those biases that may already exist in the data as a result of the human scores modeled. Our linear mixed effects regression model showed that there was a statistically significant bias. However, the marginal R^2 was found to be negligible (0.0003). In sum, we found that there was no notable difference in the model predictions for different demographic groups.

Limitations

This paper outlines our grand prize-winning approach to developing large language models (LLMs) to automatically score the ten items in the National Assessment of Educational Progress (NAEP) Math Scoring Challenge. Although the model achieved human-like agreement with the human rater score in nine out of ten items, there are some limitations to our approach. Our models were specialized to the ten items in the NAEP challenge and will likely not generalize beyond those items. New models would have to be tailored and trained for each additional item. However, while the models themselves will likely not generalize to new items, our data preprocessing and model training procedure will be able to generalize to questions outside of the dataset, provided a large enough training set. The ten items on which we trained the models were from diverse subjects within 4th and 8th grade mathematics and we expect that this process could be used internally to scale assessment solutions for new items. Assessment designers could start with limited distributions to collect enough data to train question models according to our process, then use those models to replace one or more human raters in a more general distribution.

Other limitations have to do with item design and the available training data. The items with the lowest kappa scores, including the item that did not reach the threshold of human-like performance, all had imbalanced data with very few instances from undersampled classes. More data from these classes in these items may have improved the accuracy of the models, and the importance of balanced classes needs to be considered during the item design process. Future assessment designers might use techniques from Item Response Theory to develop items that have lower difficulty (Gnaldi, 2017), ensuring a more evenly distributed response pattern for most items.

Although our models did not demonstrate bias relative to the human scores, this study says nothing about the bias of the human scorers themselves. Machine learning models can only be as good as the data they are trained on, and further research may explore bias in the training data and look for ways to reduce it. While rater bias is outside the scope of this study, rater training has been found to reduce cases of bias (İLhan, 2019; Lumley & McNamara, 1995), and is its own subject of study.

Finally, our final models were exclusively embedding-type encoder-only language models. We did experiment with open-sourced generative models including Camelidae family generative models such as Llama (Touvron et al., 2023). After hyperparameter optimization, Camelidae models for sequence classification appeared to perform nearly as well as the embedding models, but with substantially longer training times and a more complex configuration. Thus, we decided to abandon generative AI for this study and direct scoring efforts towards embedding models for predicting scores. Of note, we also experimented prompt engineering with the Vicuna model, which is also part of the Camelidae family. The Vicuna model did not result in better accuracy than the embedding models, but we had some success asking the model to explain why scores were attributed. While we were unsuccessful working with the Camelidae family of pre-trained models, generative models are a potential method of explaining score predictions in future work.

Implications

Our research indicates how imbalanced datasets with complex input can be used to develop large language scoring models for automated scoring of extended response items in mathematics. We believe that this research will support the development of automated scoring systems which will increase efficiency in terms of cost and time for scoring open response items in math assessment beyond NAEP. NCES, other assessment agencies, and math learning platforms can use these approaches to streamline scoring pipelines to reduce the resource intensive process of human scoring constructed response math items. For high stakes assessments, once models are trained on a sufficiently large dataset of human-scored responses, they can be used to replace one of the two human reviewers with an additional rater being called upon only in cases of disagreement. We hope that this increased efficiency in scoring will lead to greater adoption of constructed response items in mathematics in large standardized tests.

Including constructed response items in math assessment can have several pedagogical applications. First, constructed response can provide access to student reasoning which can give insights into math misconceptions (Landron-Rivera et al., 2018). A greater understanding of math misconceptions could be used to refine curriculum and provide personalized instruction based on student misconceptions (Nesher, 1987). Second, in low stakes assessments, constructed response models may eventually be integrated into digital math platforms to provide immediate feedback to students about the accuracy of their work allowing for increased opportunities to practice math problem solving (Hwang & Tu, 2021). Although the scope of the current work is focused on the process of training machine learning models

to provide accurate predictions of human scores of constructed response items in mathematics, we hope that improved predictions will lead to greater adoption of these types of items which will, in turn, lead to further advances in technologies to improve student math assessment and learning.

Declarations

Conflicts of Interest The authors affirm that there are no conflicts of interest.

References

Abdullah, M., Khrais, J., & Swedat, S. (2022). Transformer-based deep learning for sarcasm detection with imbalanced dataset: Resampling techniques with downsampling and augmentation. In *13th International Conference on Information and Communication Systems (ICICS)* (pp. 294–300). IEEE. https://doi.org/10.1109/ICICS55353.2022.9811196

Abercrombie, G., & Hovy, D. (2016). Putting sarcasm detection into context: The effects of class imbalance and manual labelling on supervised machine classification of twitter conversations. *Proceedings of the ACL 2016 Student Research Workshop* (pp. 107–113). Association for Computational Linguistics. https://doi.org/10.18653/v1/P16-3016

Baffour, P., Saxberg, T., & Crossley, S. (2023). Analyzing bias in large language model solutions for assisted writing feedback tools: Lessons from the feedback prize competition series. *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)* (pp. 242–246). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.bea-1.21

Baral, S., Botelho, A. F., & Erickson, J. A. (2021). *Improving Automated Scoring of Student Open Responses in Mathematics*. International Educational Data Mining Society.

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. https://doi.org/10.1162/tacl_a_00236

Bayer, M., Kaufhold, M.-A., Buchhold, B., Keller, M., Dallmeyer, J., & Reuter, C. (2023). Data augmentation in natural language processing: A novel text generation approach for long and short text classifiers. *International Journal of Machine Learning and Cybernetics, 14*(1), 135–150. https://doi.org/10.1007/s13042-022-01553-3

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

Braun, H., Kirsch, I., & Yamamoto, K. (2011). An Experimental Study of the Effects of Monetary Incentives on Performance on the 12th-Grade NAEP Reading Assessment. *Teachers College Record: The Voice of Scholarship in Education, 113*(11), 2309–2344. https://doi.org/10.1177/016146811111301101

Cochran, K., Cohn, C., Hutchins, N., Biswas, G., & Hastings, P. (2022). Improving Automated Evaluation of Formative Assessments with Text Data Augmentation. In M. M. Rodrigo, N. Matsuda, A.

I. Cristea, & V. Dimitrova (Eds.), *Artificial Intelligence in Education* (Vol. 13355, pp. 390–401). Springer International Publishing. https://doi.org/10.1007/978-3-031-11644-5_32

Crossley, S., Kyle, K., Davenport, J., & Danielle S., M. (2016). Automatic assessment of constructed response data in a chemistry tutor. *International Educational Data Mining Society*. International Conference on Educational Data Mining (EDM), Raleigh, NC. Retrieved July 16, 2024 from https://eric.ed.gov/?id=ED592642

Culpepper, S. A. (2017). The Prevalence and Implications of Slipping on Low-Stakes, Large-Scale Assessments. *Journal of Educational and Behavioral Statistics, 42*(6), 706–725. https://doi.org/10.3102/1076998617705653

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Preprint retrieved from https://arxiv.org/abs/1810.04805

Dossey, J. A., Mullis, I. V. S., & Jones, C. O. (1993). *Can students do mathematical problem solving?: Results from constructed-response questions in NAEP's 1992 mathematics assessment*. U.S. Department of Education, Office of Educational Research and Improvement.

Erickson, J. A., Botelho, A. F., McAteer, S., Varatharaj, A., & Heffernan, N. T. (2020). The automated grading of student open responses in mathematics. *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge* (pp. 615–624). Association for Computing Machinery. https://doi.org/10.1145/3375462.3375523

Fernandez, N., Ghosh, A., Liu, N., Wang, Z., Choffin, B., Baraniuk, R., & Lan, A. (2022). Automated Scoring for Reading Comprehension via In-context BERT Tuning. In M. M. Rodrigo, N. Matsuda, A. I. Cristea, & V. Dimitrova (Eds.), *Artificial Intelligence in Education* (Vol. 13355, pp. 691–697). Springer International Publishing. https://doi.org/10.1007/978-3-031-11644-5_69

Finn, B. (2015). Measuring Motivation in Low-Stakes Assessments. *ETS Research Report Series, 2015*(2), 1–17. https://doi.org/10.1002/ets2.12067

Gaye, B., Zhang, D., & Wulamu, A. (2021). Sentiment classification for employees reviews using regression vector- stochastic gradient descent classifier (RV-SGDC). *PeerJ Computer Science, 7*, e712. https://doi.org/10.7717/peerj-cs.712

Gnaldi, M. (2017). A multidimensional IRT approach for dimensionality assessment of standardised students' tests in mathematics. *Quality & Quantity, 51*(3), 1167–1182. https://doi.org/10.1007/s11135-016-0323-4

Goswami, M., & Sabata, P. (2021). Evaluation of ML-Based Sentiment Analysis Techniques with Stochastic Gradient Descent and Logistic Regression. In M. Chakraborty, R. Kr. Jha, V. E. Balas, S. N. Sur, & D. Kandar (Eds.), *Trends in Wireless Communication and Information Security* (Vol. 740, pp. 153–163). Springer Singapore. https://doi.org/10.1007/978-981-33-6393-9_17

Hancock, C. L. (1995). Implementing the Assessment Standards for School Mathematics: Enhancing Mathematics Learning with Open-Ended Questions. *The Mathematics Teacher, 88*(6), 496–499. https://doi.org/10.5951/MT.88.6.0496

He, P., Liu, X., Gao, J., & Chen, W. (2021a). DeBERTa: Decoding-enhanced BERT with disentangled attention. Preprint retrieved from https://arxiv.org/abs/2006.03654

He, P., Liu, X., Gao, J., & Chen, W. (2021b). DeBERTa: Decoding-enhanced BERT with Disentangled Attention. Preprint retrieved from https://arxiv.org/abs/2006.03654

Hogan, T. P., & Murphy, G. (2007). Recommendations for Preparing and Scoring Constructed-Response Items: What the Experts Say. *Applied Measurement in Education, 20*(4), 427–441. https://doi.org/10.1080/08957340701580736

Hwang, G.-J., & Tu, Y.-F. (2021). Roles and research trends of artificial intelligence in mathematics education: A bibliometric mapping analysis and systematic review. *Mathematics, 9*(6), 584.

İLhan, M. (2019). An Empirical Study for the Statistical Adjustment of Rater Bias. *International Journal of Assessment Tools in Education*, *6*(2), 193–201. https://doi.org/10.21449/ijate.533517

Inoue, N., & Buczynski, S. (2011). You Asked Open-Ended Questions, Now What? Understanding the Nature of Stumbling Blocks in Teaching Inquiry Lessons. *The Mathematics Educator, 20*(2), 10–23.

Ji, C. S., Rahman, T., & Yee, D. S. (2021). Mapping state proficiency standards onto the NAEP scales: Results from the 2019 NAEP reading and mathematics assessments (NCES 2021–036). *Institute of Educational Sciences, National Center for Education Statistics*. Retrieved July 16, 2024 from https://files.eric.ed.gov/fulltext/ED612877.pdf

Kang, S. H. K., McDermott, K. B., & Roediger, H. L. (2007). Test format and corrective feedback modify the effect of testing on long-term retention. *European Journal of Cognitive Psychology, 19*(4–5), 528–558. https://doi.org/10.1080/09541440601056620

Kuo, B.-C., Chen, C.-H., Yang, C.-W., & Mok, M. M. C. (2016). Cognitive diagnostic models for tests with multiple-choice and constructed-response items. *Educational Psychology, 36*(6), 1115–1133. https://doi.org/10.1080/01443410.2016.1166176

Lagakis, P., & Demetriadis, S. (2021). Automated essay scoring: A review of the field. *2021 International Conference on Computer, Information and Telecommunication Systems (CITS)* (pp. 1–6). IEEE. https://doi.org/10.1109/CITS52676.2021.9618476

Lan, A. S., Vats, D., Waters, A. E., & Baraniuk, R. G. (2015). Mathematical language processing: Automatic grading and feedback for open response mathematical questions. *Proceedings of the Second (2015) ACM Conference on Learning @ Scale* (pp. 167–176). Association for Computing Machinery. https://doi.org/10.1145/2724660.2724664

Landron-Rivera, B. A., Santiago, N. G., Santiago, A., & Vega-Riveros, J. F. (2018). Text classification of student predicate use for automatic misconception categorization. *2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1–8). IEEE. https://doi.org/10.1109/FIE.2018.8658680

Ludwig, S., Mayer, C., Hansen, C., Eilers, K., & Brandt, S. (2021). Automated Essay Scoring Using Transformer Models. *Psych, 3*(4), 897–915. https://doi.org/10.3390/psych3040056

Lumley, T., & McNamara, T. F. (1995). Rater characteristics and rater bias: Implications for training. *Language Testing, 12*(1), 54–71. https://doi.org/10.1177/026553229501200104

Ma, E. (2019). *NLP Augmentation*. Retrieved July 16, 2024 from https://github.com/makcedward/nlpaug

McCaffrey, D. F., Casabianca, J. M., Ricker-Pedley, K. L., Lawless, R. R., & Wendler, C. (2022). Best Practices for Constructed-Response Scoring. *ETS Research Report Series, 2022*(1), 1–58. https://doi.org/10.1002/ets2.12358

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. Preprint retrieved from https://arxiv.org/abs/1301.3781

Morris, W., Crossley, S., Holmes, L., Ou, C., McNamara, D., & Dascalu, M. (2023). Using Large Language Models to Provide Formative Feedback in Intelligent Textbooks. In N. Wang, G. Rebolledo-Mendez, V. Dimitrova, N. Matsuda, & O. C. Santos (Eds.), *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky* (Vol. 1831, pp. 484–489). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-36336-8_75

NAEP. (2021). *ED.gov National Assessment of Educational Progress (NAEP) Automated Scoring Challenge*. Github. Retrieved July 16, 2024 from https://github.com/NAEP-AS-Challenge/reading-prediction

NAEP. (2023). *NAEP Math Automated Scoring Challenge*. Github. Retrieved July 16, 2024 from https://github.com/NAEP-AS-Challenge/math-prediction

Nesher, P. (1987). Towards an Instructional Theory: The Role of Student's Misconceptions. *For the Learning of Mathematics, 7*(3), 33–40.

O'Neil, H. F., Abedi, J., Miyoshi, J., & Mastergeorge, A. (2005). Monetary Incentives for Low-Stakes Tests. *Educational Assessment, 10*(3), 185–208. https://doi.org/10.1207/s15326977ea1003_3

Ormerod, C. M., Malhotra, A., & Jafari, A. (2021). Automated essay scoring using efficient transformer-based language models. Preprint retrieved from https://arxiv.org/abs/2102.13136

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Peng, S., Yuan, K., Gao, L., & Tang, Z. (2021). MathBERT: A Pre-Trained Model for Mathematical Formula Understanding. Preprint retrieved from https://arxiv.org/abs/2105.00377

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1162

Phelan, J. C., Choi, K., Niemi, D., Vendlinski, T. P., Baker, E. L., & Herman, J. (2012). The effects of POWERSOURCE © assessments on middle-school students' math performance. *Assessment in Education: Principles, Policy & Practice, 19*(2), 211–230. https://doi.org/10.1080/0969594X.2010.532769

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Preprint retrieved from https://arxiv.org/abs/1910.10683

Raheja, V., Kumar, D., Koo, R., & Kang, D. (2023). CoEdIT: Text Editing by Task-Specific Instruction Tuning. Preprint retrieved from https://arxiv.org/abs/2305.09857

Raman, M., Maini, P., Kolter, J. Z., Lipton, Z. C., & Pruthi, D. (2023). Model-tuning Via Prompts Makes NLP Models Adversarially Robust. Preprint retrieved from https://arxiv.org/abs/2303.07320

Rampey, B., Dion, G. S., & Donahue, P. L. (2009). NAEP 2008: Trends in Academic Progress. NCES 2009–479. *National Center for Educational Statistics*.

Rizos, G., Hemker, K., & Schuller, B. (2019). Augment to prevent: Short-text data augmentation in deep learning for hate-speech classification. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (pp. 991–1000). Association for Computing Machinery. https://doi.org/10.1145/3357384.3358040

Rodriguez, P. U., Jafari, A., & Ormerod, C. M. (2019). Language models and Automated Essay Scoring. Preprint retrieved from https://arxiv.org/abs/1909.09482

Slepkov, A. D., & Godfrey, A. T. K. (2019). Partial Credit in Answer-Until-Correct Multiple-Choice Tests Deployed in a Classroom Setting. *Applied Measurement in Education, 32*(2), 138–150. https://doi.org/10.1080/08957347.2019.1577249

Song, K., Tan, X., Qin, T., Lu, J., & Liu, T.-Y. (2020). MPNet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems, 33*, 16857–16867.

Stedman, L. C. (2008). The NAEP long-term trend assessment: A review of its transformation, use, and findings. *Teaching, Learning, and Educational Leadership Faculty Scholarship*, *2*. Retrieved July 16, 2024 from https://orb.binghamton.edu/education_fac/2/

Sukkarieh, J., Pulman, S., & Raikes, N. (2003). Automarking: Using computational linguistics to score short free-text responses. *Proceedings of 29th International Association for Educational Assessment (IAEA) Annual Conference.* Retrieved July 16, 2024 from https://www.cs.ox.ac.uk/files/234/sukkarieh-pulman-raikes.pdf

Sukkarieh, J. Z., & Blackmore, J. (2009). *C-rater:Automatic Content Scoring for Short Constructed Responses*. Flairs Conference.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., … Scialom, T. (2023). Llama 2: Open Foundation and Fine-Tuned Chat Models. Preprint retrieved from https://arxiv.org/abs/2307.09288

Tran, N., Pierce, B., Litman, D., Correnti, R., & Matsumura, L. C. (2023). Utilizing Natural Language Processing for Automated Assessment of Classroom Discussion. In N. Wang, G. Rebolledo-Mendez, V. Dimitrova, N. Matsuda, & O. C. Santos (Eds.), *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky* (Vol. 1831, pp. 490–496). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-36336-8_76

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. Preprint retrieved fromhttps://arxiv.org/abs/1706.03762

Wang, Y., Zheng, Y., Zhu, J., & Yu, Y. (2022). LoBERTa: A composition named entity recognition method based on longformer and DeBERTa model. *International Conference on Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM)* (pp. 266–270). IEEE. https://doi.org/10.1109/MLCCIM55934.2022.00052

Whitmer, J., Deng, E., Blankenship, C., Beiting-Parrish, M., Zhang, T., & Bailey, P. (2023). Results of NAEP Reading Item Automated Scoring Data Challenge (Fall 2021). Preprint retrieved from https://osf.io/preprints/edarxiv/2hevq

Yoo, K. M., Park, D., Kang, J., Lee, S.-W., & Park, W. (2021). GPT3Mix: Leveraging Large-scale Language Models for Text Augmentation. Preprint retrieved from https://arxiv.org/abs/2104.08826

Zhang, M., Baral, S., Heffernan, N., & Lan, A. (2022). Automatic Short Math Answer Grading via In-context Meta-learning. Preprint retrieved from https://arxiv.org/abs/2205.15219

Zhang, T. (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms. *Twenty-First International Conference on Machine Learning - ICML '04* (pp. 116). Association for Computing Machinery. https://doi.org/10.1145/1015330.1015332