Automated Assessment in Math Education: A Comparative Analysis of LLMs for Open-Ended Responses

Sami Baral  
Worcester Polytechnic Institute  
sbaral@wpi.edu

Eamon Worden  
Worcester Polytechnic Institute  
elworden@wpi.edu

Wen-Chiang Lim  
Worcester Polytechnic Institute  
wlim@wpi.edu

Zhuang Luo  
Worcester Polytechnic Institute  
zluo3@wpi.edu

Christopher Santorelli  
Worcester Polytechnic Institute  
cjsantorelli@wpi.edu

Ashish Gurung  
Carnegie Mellon University  
agurung@andrew.cmu.edu

ABSTRACT
The effectiveness of feedback in enhancing learning outcomes is well documented within Educational Data Mining (EDM). Various prior research have explored methodologies to enhance the effectiveness of feedback to students in various ways. Recent developments in Large Language Models (LLMs) have extended their utility in enhancing automated feedback systems. This study aims to explore the potential of LLMs in facilitating automated feedback in math education in the form of numeric assessment scores. We examine the effectiveness of LLMs in evaluating student responses and scoring the responses by comparing 3 different models: Llama, SBERT-Canberra, and GPT4 model. The evaluation requires the model to provide a quantitative score on the student’s responses to open-ended math problems. We employ Mistral, a version of Llama catered to math, and fine-tune this model for evaluating student responses by leveraging a dataset of student responses and teacher-provided scores for middle-school math problems. A similar approach was taken for training the SBERT-Canberra model, while the GPT4 model used a zero-shot learning approach. We evaluate and compare the models’ performance in scoring accuracy. This study aims to further the ongoing development of automated assessment and feedback systems and outline potential future directions for leveraging generative LLMs in building automated feedback systems.

Keywords
Auto-Scoring, Automated Feedback, Open-End Problems, Large Language Models, Online Learning Platforms

1. INTRODUCTION
The growing integration of online learning platforms into traditional educational settings has influenced the development and direction of educational research. The global pandemic, COVID-19, resulted in the adoption of Online Learning Platforms (OLP) [1]. Consequently, various OLPs, especially in math education, have gained popularity over the recent years [15]. With the popularity of these platforms, there has been various research investigating effective teaching strategies, with many reporting on the benefit of timely and immediate feedback [5, 14, 7]. Feedback plays a crucial role in facilitating effective learning experiences, offering more than just assessments on the correctness of their answer by providing student-specific guidance. Timely feedback, in particular, can be highly effective in enabling students to rectify misunderstandings, bridge gaps in knowledge, or navigate to subsequent stages of their learning requirements. Prior exploration of effective feedback has reported on the effectiveness of feedback in enhancing learning outcomes, including the use of hints [23], explanations [18], worked-out examples [4], and common wrong answer feedback [8, 9], while others caution against the use of certain feedback designs, suggesting that poorly designed feedback can inadvertently impede student progress [9].

Automated scoring has been a focus for numerous online learning platforms, with extensive research spanning various fields, including mathematics [2], writing [20, 16], and programming [19, 22]. The initial works emphasized automating the grading of close-ended questions. However, recent advancements have extended these methodologies to include open-ended problems as well [6]. While early applications of automated scoring primarily focused on augmenting teacher resources in evaluating student responses, more recent explorations have begun to implement these techniques directly within classroom environments [17] to support students dynamically in real-time.

The recent advancement and innovation in Large Language Models (LLMs), such as ChatGPT, have introduced a transformative approach to crafting automated feedback and assessment systems within educational platforms. These developments in LLM technology have demonstrated significant potential in creating diverse mathematical content, providing support for math tutoring, offering detailed explanations, and facilitating the development of automated tutoring systems and educational chatbots that are adept at adapting to a wide range of contextual nuances.

In this study, we delve into the application of pre-trained Large Language Models (LLMs) for scoring students’ open-ended responses. We particularly assess a fine-tuned LLM
derived from Mistral—a Llama variant optimized for mathematics—and compare its efficacy with a leading non-generative model [3], currently used for the automated assessment of open-ended responses in mathematics. Additionally, we explore how these methods stack up against the capabilities of the GPT-4 model. Given the current limitations on training and fine-tuning GPT-4, we adopt a zero-shot strategy by providing the GPT-4 model with specific rubrics related to the open-ended questions. Toward this, we explore the following research questions:

1. How does an LLM fine-tuned (GOAT) with a dataset of students' responses and teacher-provided scores compare to the previous state-of-the-art, SBERT-Canberra method in predicting teacher scores for student open responses?

2. How does the pre-trained GPT4 model compare to the finetuned LLM (GOAT) in the auto-scoring task for open-ended questions?

2. METHODOLOGY

In this paper, we present a fine-tuned Large Language Model based on Llama, catered to the dataset of students' open-ended responses in mathematics. We call this model “GOAT,” which can assess a student's open-ended response in mathematics and generate a numeric score for the response. We present an in-depth analysis of this model, comparing its performance with the established method, called SBERT-Canberra, from the prior works and the conventional pre-trained GPT4 model. We talk about these methods in detail in the following subsections.

2.1 Dataset

For this study, we utilize a dataset from the ASSISTments[10] online learning platform of students’ responses to open-ended math questions with the correctness scores and feedback messages given by teachers to these open-ended responses. We selected a dataset from a specific group of about 50 teachers who used open-ended questions more frequently in their classrooms.

To train and evaluate our models, we selected 50 random open-response problems, which each had 100 student answers, feedback messages, and teacher scores. We performed an 80-20 train-test split on each question. This meant our training set included 80 student answers, feedback messages, and scores per problem for a total of 4,000 entries. We then evaluate our models on the remaining 20 students' answers, feedback messages, and scores for a total of 1,000 entries. We compare each model's score to the teacher's assigned score. We also utilize 2 math teachers to manually review 100 random test entries to determine which model performs the best. We had teachers review 2 unique entries for each of the 50 questions in our test set.

We illustrate a few examples of open-ended problems with student responses, teacher-provided feedback, and scores to these responses in Table 1. Also, Table 2 presents the distribution of teacher-provided scores within our dataset.

2.2 SBERT-Canberra

The SBERT-Canberra method from Baral et. al [2] presents a similarity-based ranking algorithm for automating assessment for open-ended responses. This method has two parts to it: i) predicting teacher score and ii) predicting teacher feedback for a given student answer. Based on the sentence-level semantic representation of students’ open-ended answers, this method presents an unsupervised learning approach. The method utilizes a historical dataset collected from an online learning platform, consisting of students’ responses with scores and textual feedback from teachers. The model compares any new student response for a math problem, with the list of responses for the same problem in the historic dataset using sentence-level embeddings from the Sentence-BERT model [21]. Using Canberra distance[13], the model finds the most similar answer from the historical dataset to any new student answer and then suggests a score and feedback based on this similar answer. This method is currently in practice in an ASSISTments learning platform, to recommend score suggestions to teachers to give to students' open-responses.

For our study, we leverage a different dataset than the prior paper, on student open-responses as described in the earlier section. We split the dataset into train and test sets, and use the training data of 50 problems to develop the SBERT-Canberra model and evaluate the results of this model on the test dataset.

2.3 GOAT

The GOAT model is our fine-tuned LLM catered to the dataset of student open responses and teacher-provided scores to these responses. To develop the GOAT model we fine-tune Mistral 7B[12]. We fine-tune based on Mistral since it has shown to beat Llama 13B on math, reading comprehension and reasoning. We fine-tuned using LoRA [11] since it uses less GPU memory and time and avoids catastrophic forgetting.

To acquire input-output pairs for fine-tuning, we utilize the illustrative grading rubric to design an instructional prompt for each pair, as shown in Figure 1, amalgamating a math problem and a student’s answer into the input, while treating a real teacher’s score as the desired output. We utilized 4000 entries data in the training split for fine-tuning and 1000 entries for testing. Fine-tuning spans 4 epochs with 10 warm-up steps. We initialize the learning rate to 0.0002 and apply a cosine annealing schedule. To address memory constraints, we adopt the gradient accumulation technique, setting gradient accumulation steps to 2, partitioned into micro-batches of 2. The training process, conducted on a single A100 GPU, lasts approximately 2 hours and yields a near-zero loss function when complete.

We determined the optimal inferencing hyperparameters using a validation set of 100 entries which was a subset of the train set. We found argmax, by finding the parameters which minimized the MSE of our score compared to the teacher score. We found argmax, to be temperature set to 0.5, top_p to 0.5, and top_k to 30.

2.4 GPT4
Table 1: Examples of student open-responses with, teacher-provided feedback and scores to these answers taken from our dataset.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Student Answer</th>
<th>Teacher Feedback</th>
<th>Teacher Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explain why 6:4 and 18:8 are not equivalent ratios.</td>
<td>You cannot multiply 4 into 6 and you cannot multiply 8 into 18.</td>
<td>I somewhat see what you are doing but instead you need to see how do you get from 6 to 18 and is that the same scale factor to get 4 to 8.</td>
<td>1</td>
</tr>
<tr>
<td>Explain why 6:4 and 18:8 are not equivalent ratios.</td>
<td>They are not equivalent ratios because 6 went into 18, 3 times and 4 went into 8, 2 times</td>
<td>Great job!</td>
<td>4</td>
</tr>
<tr>
<td>Write an equation that represents each description. The opposite of negative seven</td>
<td>–7 = 7</td>
<td>Great job!</td>
<td>4</td>
</tr>
<tr>
<td>Write an equation that represents each description. The opposite of negative seven</td>
<td>7</td>
<td>Can you write an equation?</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 2: Score Distribution

<table>
<thead>
<tr>
<th>Score</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Responses</td>
<td>771</td>
<td>768</td>
<td>1086</td>
<td>816</td>
<td>1559</td>
</tr>
</tbody>
</table>

Figure 1: The fine-tuning process for the GOAT model for the downstream task of predicting teacher score and feedback for student open-responses in mathematics.

GPT4 is the state-of-the-art language model developed by OpenAI, designed to understand and generate human-like text based on the prompts provided as input to the model. This model has been retained on a diverse and extensive dataset of texts available from the internet, books, and other sources. GPT4 has proven to have significantly improved capabilities in terms of understanding the context, generating relevant text, and handling various complex language tasks. As such in this work, we explore the applicability of this pre-trained language model in predicting a score and generating appropriate feedback for students’ open-response answers in mathematics.

For our method, we employ the “GPT4 Turbo” model, which is the optimized version of the GPT4 model designed to enhance the efficiency and response time and maintain the high quality of the original GPT4 model. With the real-world applicability of this model, being cost and time-efficient, we focus on the use of this version of GPT4 for our study. To explore the performance of the pre-trained model in math assessment tasks, we employ a zero-shot learning approach with GPT-4, where we do not provide any context examples to the model. For this, we follow a carefully designed prompting strategy, where we provide the model with the
Among the three models, the GOAT model outperformed the specific assessment of their predictive capabilities. For the prompt engineering process, we followed an iterative approach involving two researchers in math education, and our prompt for the study is shown in Figure 3.

Table 3: Final Prompt used as input for GPT-4 model to generate score and feedback for student open-ended responses in mathematics.

You are a middle school math teacher, giving helpful feedback to students on their mathematical reasoning on open-response questions. Keep your feedback direct, under 50 words, and do not give away the answer in your feedback.

<table>
<thead>
<tr>
<th>Problem:</th>
</tr>
</thead>
<tbody>
<tr>
<td>{body}</td>
</tr>
<tr>
<td>Student's Answer:</td>
</tr>
<tr>
<td>{value}</td>
</tr>
<tr>
<td>Scoring Rubric:</td>
</tr>
<tr>
<td>1 Students should get 4 points if their work is complete and correct, with complete explanation or justification.</td>
</tr>
<tr>
<td>2 Students should get 3 points if their work shows good conceptual understanding and mastery, with either minor errors or correct work with insufficient explanation or justification.</td>
</tr>
<tr>
<td>3 Students should get 2 points if their work shows a developing but incomplete conceptual understanding, with significant errors.</td>
</tr>
<tr>
<td>4 Students should get 1 point if their work includes major errors or omissions that demonstrate a lack of conceptual understanding and mastery.</td>
</tr>
<tr>
<td>5 Students should get 0 points if they do not attempt the problem at all.</td>
</tr>
</tbody>
</table>

Table 4: Model Performances on Scoring

<table>
<thead>
<tr>
<th>Model</th>
<th>AUC</th>
<th>RMSE</th>
<th>Kappa</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBERT</td>
<td>0.662</td>
<td>1.364</td>
<td>0.362</td>
</tr>
<tr>
<td>GOAT</td>
<td>0.697</td>
<td>1.119</td>
<td>0.422</td>
</tr>
<tr>
<td>GPT-4</td>
<td>0.639</td>
<td>1.16</td>
<td>0.266</td>
</tr>
</tbody>
</table>

4. DISCUSSION AND CONCLUSION

In this paper, we introduce the GOAT model, a fine-tuned bespoke solution designed for predicting scores for student responses to open-ended math questions. Our results demonstrate that GOAT outperforms the previous benchmark set by the SBERT-Canberra model in the auto-scoring domain across all three evaluation metrics: AUC, RMSE, and Kappa. However, the model’s accuracy, while fair, signals the need for further refinements before full-scale deployment is feasible.

Notably, both GOAT and SBERT outperformed the GPT-4 model in the scoring task. However, it’s crucial to recognize that GPT-4 serves as a generic pre-trained model and hasn’t undergone any task-specific fine-tuning or training that both SBERT and GOAT have undergone by utilizing teacher grades on student responses. This distinction is underscored by the alignment between the scoring patterns of SBERT and GOAT with those of the teachers, particularly in their propensity to award scores of 4. In contrast, GPT-4’s achieved a distinct score distribution, as illustrated in Figure 2. This variance highlights the nuanced differences in model training and the potential impact on their scoring capabilities. Given that the GPT-4 model utilized a grading rubric from Illustrative Math to assess the quality of student responses, future research should delve into the specific factors contributing to the differences in grading outcomes between GPT-4 and the other models. Identifying the root cause of this grading discrepancy is essential. It
could indicate whether the variance is due to teachers’ leniency stemming from personalization, a misalignment between the rubric’s literal interpretation and teacher expectations in practice, or perhaps a combination of both factors.

![Figure 2: Score Distribution of Teachers compared to the predictions from the three models of SBERT, GOAT and GPT-4 across the test dataset used for the study.](image)

Overall, we present a fine-tuned GOAT model to evaluate a student’s open-ended answer and generate a score. Comparing this method with the traditional method of automated assessment – the SBERT-Canberra method and the conventional pre-trained GPT-4 model, we find that this method outperforms both the models in the autoscoring task. However, for the feedback generation, the conventional GPT-4 model beats the other two when evaluated by human raters with prior teaching experience.

5. ACKNOWLEDGMENTS

We would like to thank past and current including NSF (2118725, 2118904, 1950683, 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, 1535428), IES (R305N210049, R305D210031, R305A170137, R305A170243, R305A190401, R305A120125, R305R220012, R305T240029), GAANN (P200A120238, P200A180088, P200A150306), EIR (U411B190024 S411B210024, S411B220024), ONR (N00014-18-1-2768), NIH (via a SBIR R44GM146483), Schmidt Futures, BMGF, CZI, Arnold, Hewlett and a $180,000 anonymous donation. None of the opinions expressed here are those of the funders.

6. ADDITIONAL AUTHORS

Additional authors: Neil Heffernan (Worcester Polytechnic Institute, email: nth@wpi.edu).

7. REFERENCES


