Evaluating the Potential of LLMs for Better Short Answer Scoring

Aleksandar Todorov, Elisa Klunder and Julia Eva Belloni
Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, The Netherlands
{*a.todorov.4, e.klunder.1, j.e.belloni*}*@student.rug.nl*

Keywords: LLMs, Automated Short-Answer Grading, Fine-Tuning, Education.

Abstract: Automated Short Answer Grading (ASAG) has emerged as a promising tool for the challenge of assessing open student responses in an efficient and scalable manner as manual grading of such open short answers is labor-intensive and time-consuming. In this study, we present several ways of refining LLMs to fit the task of grading student short-answer responses robustly, fairly, and consistently, including a task-specific approach and a combined variant, being able to assess different tasks within the same model. In this regard, we explore two key questions: (1) Are transformer-based models suitable for short-answer grading? (2) Can a single transformer-based model effectively generalize across diverse tasks? The experimental results showed the significant potential of fine-tuned LLMs in ASAG. We further compared different fine-tuning strategies and the experimental results showed that full-fine-tuned models outperformed other fine-tuning approaches.

1 INTRODUCTION

Essays and short-form answers have been widely used as a tool to evaluate students across educational levels. Compared to multiple-choice questions, short-form answers require a deeper understanding of the material and a capacity for coherent expression. However, grading these responses manually is time-consuming, leading educators to favor multiple-choice questions, which often provide only a superficial measure of a student's knowledge despite their ease of grading. As online courses gain traction and education gets more accessible to large parts of the population, the demand for a fair, fast, and scalable solution for automated grading of textual answers has entered the spotlight. Automated Short Answer Grading (ASAG) seeks to meet this need by enabling fast and reliable assessment using the newest available technologies and computational resources in a systematic way.

The use of computational methods for grading written responses has a long history, originating with the 1966 pioneering work of Page (1966), which introduced an automated essay scoring system called Project Essay Grade (PEG). Since then, automatic grading of natural language responses has evolved into a substantial field of study, with increasingly complex models being developed, and Machine Learning (ML) techniques gaining significant traction. Nonetheless, most models kept relying on similar hand-engineering surface-level features (Galhardi and Brancher, 2018) and thus failed to capture the context and meaning of student responses. Furthermore, the contribution of each of these features to the final grade is usually assumed to be true based on preconceived notions without accounting for deeper nuances in writing (for example, valuing longer responses without considering conciseness and clarity).

Recently, a new exciting method has gathered considerable attention in Natural Language Processing (NLP) due to its transformative potential: Large Language Models (LLMs) (Ding et al., 2023; Zhao et al., 2023). Bonner et al. (2023) emphasized the transformative potential of LLMs in education. In particular, the authors point out how LLMs are well-suited to help in educational tasks like text evaluation because of their ability to understand and generate coherent text. For this reason, we believe LLMs could constitute a game-changer in the field of ASAG: not only could they achieve high agreement with human scorers, but they also simplify the grading process for educators by eliminating the need for manual feature engineering and enabling models to learn directly from the text without any additional manipulation. This approach could be highly versatile: merely providing the text would enable the model to automatically capture relevant features and patterns.

Additionally, because LLMs are pre-trained on vast amounts of real-world textual data, they possess a remarkable capability to generalize across various

Todorov, A., Klunder, E. and Belloni, J. E.
Evaluating the Potential of LLMs for Better Short Answer Scoring.
DOI: 10.5220/0013291700003932

In *Proceedings of the 17th International Conference on Computer Supported Education (CSEDU 2025) - Volume 2*, pages 108-119
ISBN: 978-989-758-746-7; ISSN: 2184-5026

tasks and exam questions without requiring extensive retraining for each new task (Reizinger et al., 2024). Recent studies indicate that transformer-based models fine-tuned with instructional data exhibit impressive generalization to unseen tasks (Zhang et al., 2023b). By training across multiple tasks, these models develop an understanding of diverse input types, learning underlying representations that distinguish well-written answers from poorly phrased ones without altering the model's core architecture. While task diversity and unique requirements can present challenges, instruction-based fine-tuning helps address these complexities, further enhancing model adaptability (Zhang et al., 2024). This flexibility makes LLMs particularly valuable for short-answer scoring.

This paper aims to investigate two fundamental questions: (1) Are transformer-based models suitable for short-answer grading? (2) Can a single transformer model generalize effectively across diverse tasks? By leveraging state-of-the-art LLMs, we seek to create a reliable and robust approach to evaluating student answers, examining both the appropriateness of transformers for scoring and their potential for task generalization. We believe this can advance the discussion on accessible and efficient assessment tools for educators, particularly in settings like Massive Open Online Courses (MOOCs), which can be used for bridging the knowledge gap in settings where public education is not easily accessible. LLMs offer a powerful solution to meet these demands, enabling large-scale assessments without sacrificing accuracy or fairness.

The remaining part of this paper is comprised of a literature review, illustrating both a historical perspective and an analysis of the state-of-the-art models for ASAG. Next, we present the methods, techniques, and methodologies used in this research to ensure reproducibility. The data, experiments, and evaluation procedures are detailed, followed by the results and an in-depth analysis of the model's performance. Finally, we conclude by summarizing our findings, emphasizing key achievements, and acknowledging the primary limitations and future directions of the research.

2 RELATED WORK

In the sixty years following the work from Page (1966), the field of Automated Short Answer Grading has evolved significantly, from early feature-engineered models to advanced LLMs. This section aims to provide a historical perspective on the field and an overview of the most relevant currently existing literature on the topic.

2.1 Feature-Engineered Models

In the early stages of research, hand-engineered features along with traditional machine learning models were the primary methods for assessing short-answer responses. These approaches relied heavily on previous research in general linguistics and natural language processing. A prime example for the use of hand-engineered features like grammar, vocabulary, and average word length, was proposed by Attali and Burstein (2006). This approach provided the advantage of transparency and explainability of the scores. However, it lacked depth in evaluating content, as it primarily measured surface characteristics without accounting for contextual meaning.

To address some of these limitations, Mohler et al. (2011) introduced a combination of the Bag-of-Words (BoW) approach with dependency graphs. While BoW represents data based on word frequency, it struggles with capturing polysemous word meanings in context, leading to challenges in interpreting content accurately. This approach, though effective for some tasks, lacked the semantic depth needed for deeper text understanding.

Building on all these approaches, Kumar et al. (2020) introduced *AutoSAS*, an efficient machine learning method for short answer scoring (SAS) based on clever feature engineering and a random forest architecture. Trained on the popular ASAP-SAS benchmark dataset (Barbara et al., 2012), AutoSAS introduced a new set of features, including lexical diversity, Doc2Vec embeddings, and prompt-content overlap, to capture content-based similarities between student responses and ideal answers. This model demonstrated a notable improvement in prediction accuracy and scalability across diverse domains, establishing itself as the state-of-the-art feature-engineered model for the ASAP-SAS dataset. Since this dataset will also be used in our study, the work by Kumar et al. (2020) will serve as an important benchmark for assessing the impact of shifting from hand-engineered features towards more complex pre-trained models like LLMs in the field of ASAG.

2.2 Recurrent Neural Networks

The advent of neural networks, particularly Recurrent Neural Networks (RNNs), marked a major shift in ASAG research, as new algorithms allowed the modeling of sequential dependencies, such as in-between sentence relations. Taghipour and Ng (2016) leveraged Long Short-Term Memory (LSTM) networks to build an end-to-end system for essay scoring that not only demonstrated near human-level accuracy but also bypassed the need for hand-crafted features by directly learning from raw text input. Similarly, Saha et al. (2019); Tashu et al. (2022) used Bi-directional Long-Short Term Memory (BiLSTM) networks to encode answers into dense feature representations and model long-term dependencies. These methods were made possible by the internal hidden state of recurrent neural networks, which enables modeling of underlying latent variables and thus preserves important context from the previous text. However, the inability to parallelize computations of these models and their inherent instability caused by vanishing and exploding gradients hampered their scalability and thus wider adoption, especially as more research projects and companies started gaining access to better GPUs.

2.3 Transformer-Based Models

Transformers (Vaswani et al., 2023) are a type of neural network architecture, mostly suited for LLMs, and are designed to process and generate text by modeling probability distributions. These distributions enable the model to predict the next token (typically words or parts of words) based on the context. For instance, when generating a sentence, the model evaluates the probability of each possible word at a given position by using self-attention mechanisms, allowing the model to weigh all preceding tokens defined within a span, capturing contextual relationships. The token with the highest probability is then selected, and this process continues iteratively until a special end-of-sequence token is generated, indicating the conclusion of the prediction. Developing transformer models requires extensive pre-training on large datasets, enabling it to learn linguistic patterns, syntax, and semantics.

In recent years, transformer-based LLMs have shaped the field of Natural Language Processing. Although these architectures excel at processing and generating human-like language, they may not initially appear well-suited to tasks like short answer scoring, where the focus is on accuracy, precise and reliable outputs, and interpretability, as opposed to the open-ended generation typical of traditional LLM applications. However, with fine-tuning techniques and the ability to constrain outputs effectively, LLMs have proven highly effective for ASAG tasks.

BERT (Sung et al., 2019), for instance, demonstrated the potential of bidirectional context modeling for short answer scoring, achieving a notable improvement in performance over earlier methods. This research demonstrated that transformers, with their ability to generate contextual embeddings through self-attention mechanisms, significantly improved the performance of short answer grading models. However, BERT's reliance on domain-specific fine-tuning remained a key limitation of this study, as its performance tended to drop significantly when applied to different domains without additional training.

Building on BERT's foundations, research by Latif and Zhai (2024) evaluated GPT-3.5 for ASAG, noting its flexibility and higher performance in multi-label and multi-class grading tasks. However, the study also noted concerns regarding the potential biases that LLMs like GPT-3.5 might inherit from their (undisclosed) training data. In a field like automated grading – where outcomes can have real and long-lasting impacts on individuals' academic and professional lives, such potential biases are hardly acceptable, regardless of performance improvements.

While all previous approaches involved fine-tuning LLMs to fit specific datasets, research by Kortemeyer (2023) evaluated GPT-4's performance for ASAG tasks without fine-tuning. This research showed that, although GPT-4 could match the performance of some older hand-engineered systems, it struggled with nuanced grading scenarios and did not reach the accuracy levels of fine-tuned BERT models. This finding demonstrates that, despite the impressive advancements in large pre-trained LLMs capable of tackling diverse tasks, fine-tuning remains essential to achieving the necessary accuracy and alignment with human graders in ASAG.

Finally, recent developments in zero-shot learning have highlighted the potential of LLMs to generalize to multiple and even unseen tasks. Wei et al. (2021) introduced instruction fine-tuning to enhance zero-shot learning across multiple tasks. The main idea was to treat each question-answer dataset (task) as an instruction that was passed to the LLM, from which the model learned to generalize to new tasks. The model's performance was compared with other models, including GPT-3, and the conclusion reached was that instruction fine-tuning significantly improved the model's ability to do well in unseen tasks. Moreover, instruction prompting remains an effective technique that could sometimes achieve impressive results even for short and simple prompts (Zhang et al., 2024). For this reason, we believe that instruction fine-tuning could enhance model performance and generalization and will therefore be a key component in the conducted experiments and analyses. These studies collectively emphasize the promise of LLMs for ASAG but also highlight challenges, such as interpretability and domain adaptation. This body of research forms the foundation for our paper, which seeks to leverage LLMs' strengths in both contextual understanding and generalization to develop a scalable, fair, and efficient solution for automated short-answer grading.

3 METHODS

This study explores the effectiveness of different fine-tuning strategies for LLMs in automated short-answer grading. Our objectives include selecting a model and evaluating the model's task suitability, examining multi-task learning potential, testing the impact of instruction prompting, and improving computational efficiency. The techniques used to refine the pre-trained model to fit the domain-specific task of short-answer scoring are full fine-tuning, LoRA fine-tuning, and Prompting.

3.1 Model Selection

With the defined multi-label classification task, we finetune a Bidirectional Encoder Representation from Transformers (BERT) (Devlin et al., 2018) and conduct our experiments. This model is suited for the task, since its bidirectional encoder-only architecture focuses primarily on contextual encoding and textual comprehension. Moreover, empirically it has already obtained relevant results in the field of short-answer scoring (Haller et al., 2022). BERT has been shown to achieve state-of-the-art performance on a different benchmarking dataset by Sung et al. (2019).

3.2 Full Fine-Tuning

Fine-tuning is a transfer learning technique often applied to large language models in order to adapt them for a specific task (Zhang et al., 2023a). The process involves using a pre-trained source model, in this case, BERT, and modifying it so that the output layer is replaced by a new one to match the short-answer grading task. The target model is trained on the dataset, described further in Section 4.1, with the output layer trained from scratch while the other layers are only adjusted to adapt to the specific task. In this case, fine-tuning is applied to the multi-label classification problem, using categorical cross-entropy loss to optimize the model.

3.3 LoRA Fine-Tuning

LoRA (Low-Rank Adaptation) is a parameter-efficient fine-tuning technique that reduces the number of trainable parameters, enabling faster and more resource-efficient training. Instead of updating the entire weight matrix of the pre-trained model, two smaller update matrices, obtained through low-rank, are trained on the new data, while the original weight matrix is kept frozen. The final outputs are obtained through a combination of the original weight matrix and the update matrices. Notably, LoRA is less sample efficient compared to fine-tuning and was shown to substantially underperform full fine-tuning (Biderman et al., 2024), thereby, in tasks requiring reliable performance full fine-tuning is to be preferred. Additionally, LoRA has a more gradual training process but might take longer to achieve comparable results to full fine-tuning, even though each epoch takes longer to complete. The LoRA fine-tuning strategy further introduces several hyperparameters that might not be highly suitable for the automated grading task as it adds another layer of complexity. This point is expanded in Section 5, after a thorough analysis of the achieved performance.

3.4 Prompt Engineering

Prompt engineering is used to enhance the performance of the model by giving it a wider context. In this case, an instruction-prompting technique was implemented, by directly informing the model about the goal it needs to accomplish. It is expected that this approach will allow the model to learn faster and understand the objective it needs to perform. A sample model's prompt is:

Grade this student's answer on a scale `[scale]`, focusing on grammar, lexical variability, and task relevance.,

where `[scale]` is either 0-2 or 0-3, depending on each specific essay set.

4 DATA AND EXPERIMENTS

4.1 Data

The dataset for Automated Student Assessment Prize Short Answer Scoring (ASAP-SAS) by Barbara et al. (2012) was released as part of a Kaggle competition in 2013, sponsored by the Hewlett Foundation. It contains more than 20,000 short-answer responses on 10 different tasks mainly from Grade 10 students from the United States. Each topic is based on either Science, Arts, Biology, or English. All the answers have been hand-graded on a scale of 0-2 or 0-3 (specified in each task description document) and double-scored for reliability, however, the second score has no effect on the final score received. Some of the task descriptions can be found in Table 1. The data has been split by the authors into a train set, consisting of 17,044 entries, and a test set, composed of 5225 responses.The tasks vary across domains, which is intended to test the capabilities of the developed model(s). For instance, task 1 regards scientific reasoning. In this task, students are presented with a partial scientific experiment and need to describe what additional information is needed to replicate it. The purpose of this task is to test students' scientific understanding and logical reasoning about experimental settings.

The other nine tasks follow a similar structure, requiring short responses (an average of 50 words per answer) on a topic demanding reasoning and critical thinking. Few of the tasks involve referring to external sources.

Table 1: Some of the tasks in the dataset and their descriptions.

Topic	Task
Biology	List and describe three processes used by cells to control the movement of substances across the cell membrane.
English	Read the article and explain how the author organizes it. Support your response with details from the article.
Arts	Read the article and explain how pandas in China are similar to koalas in Australia and how they both are different from pythons. Support your response with information from the article.

All the data is in a text format, but four of the ten datasets have been manually transcribed and might contain typing errors. However, an addition of such noise might be beneficial as it simulates the real-world setting, where textual inconsistencies often occur.

4.2 Evaluation Method

4.2.1 Baselines

As baselines for evaluation, we use the pre-trained BERT model from HuggingFace and the state-of-the-art feature-based random-forest model proposed by Kumar et al. (2020). In the former case, we contrast whether our fine-tuning and prompt engineering approach will improve the performance and feedback-providing capabilities of the untuned LLM. Moreover, the BERT baseline is evaluated both with and without the instruction-engineering technique mentioned previously. For the latter, we compare only the performance in terms of the κ score across all essay tasks, defined in the section below.

4.2.2 Cross-Validation and Early Stopping

The train was further split into a training set (80%) and validation set (20%) for all experiments using a random seed 4242 to ensure reproducibility. Hence, the train set contained 13635 data points, the validation set 3409 data points, and the test set 5225, distributed across all 10 essay tasks.

The data was shuffled before splitting, to ensure similar distributions of essay sets. Furthermore, it reduces the potential effect of primacy and recency biases during the training process. The validation set was used for the early-stopping mechanism. If the model is trained for too many epochs, it may eventually overfit by memorizing the labels or the noise without learning the underlying patterns, leading to poor generalization on new data sets. This system monitors the validation loss and halts training if no improvement is observed over a specified number of epochs, defined by the `patience` parameter (see the Experiment Details section). When this limit is reached, the model detects the potential for overfitting, the training is stopped and the model with the lowest validation loss is saved.

4.2.3 Evaluation Metric

Barbara et al. (2012) provide a suggested metric for evaluating model performance on the dataset, namely the Quadratic Weight Kappa (QWK). This metric compares the agreement between the predicted scores of our model and the ones provided by human graders. It also accounts for the probability of the two scores being equal by chance. The QWK evaluates to 1 when the predicted value and the known one are the same and 0 when there is an agreement by random. A negative QWK can be received if the agreement is below what is expected due to random chance. A batch of student responses has N possible scores, and two raters, *Rater A* (human) and *Rater B* (model).

Each short-answer response has been scored as a tuple (i, j), where i is the human score and j is the model-generated score. An $N \times N$ histogram matrix O is calculated over the ratings of the responses, such that each $O_{i,j}$ corresponds to the number of answers that received rating i by *Rater A* and rating j by *Rater B*. A weight matrix W measures the disagreement between any two scores.

$$w_{i,j} = \frac{(i-j)^2}{(N-1)^2},$$

with i, j respectively being the human and model score, and N the number of scores possible. Moreover, an expectation matrix

$$E = e_a \otimes e_b$$

is calculated, where e_a and e_b are the histogram vectors of the human values and the model predicted values, respectively and $\otimes$ denotes the outer product. In a histogram vector, each entry represents the frequency of a certain score in the data. In this case, there are N possible scores, so at each index the count of a score is registered. The QWK metric κ is finally calculated as

$$\kappa = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}}.$$

With this score, we evaluate both the baseline model and the fine-tuned models. The mean of the κ scores is taken over all responses from the 10 different tasks, which is consistent with the approach of Kumar et al. (2020). In the field of short-answer scoring, an automated system is considered to be acceptable if it achieves a QWK score of at least 0.70 (Doewes et al., 2023). An interpretation of κ values can be seen in Table 2.

Table 2: Quadratic Weighted Kappa Interpretation Scale. Table from Doewes et al. (2023).

κ	**Interpretation**
< 0	Less than chance agreement
0.01 - 0.20	Slight agreement
0.21 - 0.40	Fair agreement
0.41 - 0.60	Moderate agreement
0.61 - 0.80	Substantial agreement
0.81 - 1.00	Almost perfect agreement

4.3 Experimental Details

The experiment was designed based on the dataset, which is organized into ten independent tasks (or essay sets). The configuration was made around the following three main conditions:

- A separate model (called *task-specific*) for each essay set or a model for all ten tasks at once (referred to as *combined*).
- Providing or not providing the model(s) with an instruction prompt. Those model configurations are also referred to as *instructed* and *uninstructed*.
- Performing full fine-tuning or parameter-efficient fine-tuning using Low-Rank Adaptation (LoRA).

These options resulted in eight distinct experiments. The motivations behind each experiment are given in the next sections. Each experiment was executed systematically to maintain consistency and ensure reliable results. The common goal across all experiments is to predict the score of a student's response using the efficacy of LLMs in considering textual dependencies and context. We hypothesize that our approach will yield comparable results to traditional machine learning models, eliminating the need for labor-intensive feature engineering, while allowing both task-specificity (singular models per essay set) and task-generality (combined model for all essay sets).

4.3.1 Full Fine-Tuning Experiments

The first set of experiments focuses on full fine-tuning. We further compare the performance of task-specific models, where a separate BERT model is fine-tuned for each of the essay tasks, and a combined model, fine-tuned for all of the essay sets at once. Moreover, we examine the effect of instruction-prompting, by adding another variable to the configuration, i.e., whether the model is prompted with the instruction or not.

In the first experiment, a distinct BERT model was trained for each essay task, allowing it to learn the specific grading criteria relevant to that task, without the use of an instruction prompt. The second experiment employs a similar strategy, however, a single BERT (task-specific) model is fine-tuned across all ten essay sets simultaneously, without instruction prompts. This approach aimed to exploit potential common mistakes and criteria between the tasks to enhance generalization capabilities. The task-specific and the combined experiment strategies can be easily differentiated by examining Figures 1 and 2, respectively.

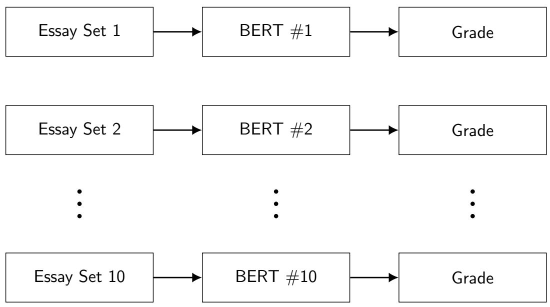

Figure 1: The task-specific experiment strategy, where a single BERT model is fine-tuned for each of the 10 essay sets in the dataset.

The third and fourth experiments involve the addition of instruction prompting. The third experiment is equivalent to the first experiment, except that it was fine-tuned with the aforementioned instruction, guid-

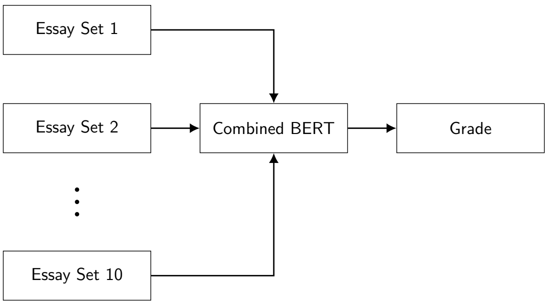

Figure 2: The combined BERT experiment strategy, where one BERT model is fine-tuned for all the 10 essay sets at once, resulting in a more general model.

ing the model in understanding what is expected from it. Similarly, the fourth experiment is equivalent to the second experiment, with the addition of the instruction prompt in every input.

All of the hyperparameters are in accordance with the `transformers` library, which was used extensively in this research for the fine-tuning and pre-trained models. The hyperparameters used for the first four fine-tuning experiments were:

- `batch_size`: Training in batches of 32 proved to be the most computationally appealing and significantly faster while retaining performance. Moreover, there are recommendations that the batch size should not be tuned to directly improve validation set performance (Godbole et al., 2023) and there is currently no solid evidence that the batch size affects the maximum achievable validation performance (Shallue et al., 2018). Hence, the batch size was not further tuned and was left as 32 for faster computation.
- `epochs`: The epochs for the full fine-tuning experiments were set to 10. Ideally, we wish to specify slightly more epochs than needed, the training does not finish but rather the early-stopping mechanism is triggered, as a means of regularization.
- `patience`: The hyperparameter for the early-stopping mechanism. This value was set to 2, as empirically, models trained rather quickly (1-2 epochs), after which validation loss started to decline.
- `weight_decay`: Models in `HuggingFace` are fine-tuned through the `AdamW` optimizer (Loshchilov and Hutter, 2019), which allows for the weight decay of all layers (except bias and `LayerNorm`), as a further form of regularization. This value is typically 0.01, which was also used in those experiments.

4.3.2 LoRA Fine-Tuning Experiments

Another set of four experiments was conducted using LoRA fine-tuning. The experiments are precisely equivalent to the full fine-tuning ones, except that a LoRA configuration grid was used as a means of parameter-efficient fine-tuning. The specified hyperparameters for this set of experiments was as follows:

- `batch_size`: 32, equivalent to the full fine-tuning experiments.
- `epochs`: LoRA fine-tuning took more epochs until the early-stopping mechanism took effect (20-30), hence the number of epochs was increased to 50 so that it could be triggered effectively.
- `patience`: 2, equivalent to the full fine-tuning experiments.
- `weight_decay`: 0.01, equivalent to the full fine-tuning experiments.
- `r`: The rank of the matrices in the adaptation layers, which determines the number of parameters, was tested with values of 8, 16, 32, and 64. A rank of 16 was selected as the optimal setting.
- `lora_alpha`: The scaling factor for the output of the reduced rank matrices from LoRA. The original LoRA paper (Hu et al., 2021) recommends a fixed value of 16, instead of treating it as a tunable hyperparameter, hence was chosen as so.
- `target_modules`: The targeted modules for the `uncased-base-BERT` were the attention layers, namely `query` and `value`.
- `lora_dropout`: The dropout probability for the LoRA layers, introducing a further regularization effect. During training, it randomly sets the elements of an input tensor to 0, which helps prevent overfitting by decorrelating the activations of the neurons in the network.

A summary of the experiment configuration can be seen in Table 3.

Table 3: Experiment Configuration Grid: Fine-Tuning Approaches, Instruction-Prompting Techniques, and Model Types.

№	Fine-Tuning	Instruction	Model Type
1	Full	No	Task-Specific
2	Full	No	Combined
3	Full	Yes	Task-Specific
4	Full	Yes	Combined
5	LoRA	No	Task-Specific
6	LoRA	No	Combined
7	LoRA	Yes	Task-Specific
8	LoRA	Yes	Combined

5 RESULTS AND DISCUSSION

5.1 Results

The results of the eight conducted experiments are reported in Table 4. System metrics are further presented in Table 5. The next sections present a comprehensive analysis of those results.

5.2 Qualitative Analysis

Considering the number of experiments conducted, multiple subsections address the targeted comparisons among models and experiment configurations.

5.2.1 BERT Baseline

The experimental results showed that both the instructed and uninstructed pre-trained BERT baseline received a QWK of 0, meaning that their predictions were made completely at random.

All fully fine-tuned models achieved performance significantly better than the BERT baseline across all essay sets. Similarly, LoRA models exceeded the BERT baseline, except for the LoRA, instructed, task-specific model in sets 1, 4, and 9, and the LoRA uninstructed, task-specific in sets 4 and 7. In those cases, the predictions were completely at random ($\kappa = 0$) or there was a disagreement between the actual grades received and the predicted scores ($\kappa < 0$).

The fully fine-tuned models mostly satisfy the criterion in the deployment of automated grading systems of $\kappa > 0.7$. We consider those results to be sufficient for the beginning of this novel approach.

5.2.2 Kumar et al. (2020) Baseline

The random forest model used by Kumar et al. (2020), which was heavily hand-engineered to fit the domain of short-answer grading, outperformed most of the models in this study. However, the Full, Uninstructed, Task-Specific model achieves the same QWK score of $\kappa = 0.62$ in essay set 8, while the two fully fine-tuned combined models outperform the state-of-the-art with $\kappa = 0.66$ and $\kappa = 0.67$.

Notably, most fine-tuned models differed from the random forest model by no more than 0.10 on any essay set. Hence, we consider the LLM approach to be comparable with the state-of-the-art machine learning method. According to Table 2, all task-specific and combined models fall within moderate, substantial, or almost perfect agreement with the human graders.

The LoRA models significantly underperformed the full fine-tuned models, and hence the random forest model. A further comparison of the LoRA models is given in one of the next subsections. Nonetheless, all LoRA models at most display a slight agreement with the actual scores.

5.2.3 Task-Specific vs. Combined BERT

Within fully fine-tuned models, minimal differences were observed. The two most notable differences can be noticed if we compare the task-specific versus combined models in set 10 for the uninstructed models and set 8 for the instructed models. In both these specific cases, the combined models perform better than the task-specific ones by 0.08 κ. In general, the combined BERT models exhibit reliable performance and generalizability across tasks. Among LoRA models, the combined architecture generally performed better, whereas the task-specific model type displayed slight or random agreement with the real scores.

5.2.4 Uninstructed vs. Instructed Models

Overall, minimal performance differences were showcased in the uninstructed vs. instructed configuration. Notably, fully fine-tuned task-specific models benefited on essay sets 5, 6, and 10, improving by 0.19, 0.10, and 0.07 in terms of the κ score. Even so, the general effect of instruction-prompting in this study remains inconclusive. This is further addressed in discussion and limitations.

For the LoRA models, adding the instruction reduced performance, compared to their uninstructed counterparts. Specifically, including the instruction in the prompts for the combined models lowered the κ score in sets 9 and 10 by 0.24 and 0.23, respectively. This substantial difference suggests that the instruction prompt made the ASAG task significantly harder for the LoRA models.

5.2.5 LoRA vs. Full Fine-Tuning

The full fine-tuning configuration significantly outperformed LoRA models across all essay sets and settings. Moreover, LoRA models took more than half of the time to train, even though they are designed for computational efficiency (see System Metrics section). We hence consider LoRA models not suitable for the automated grading task. This is further elaborated on in the discussion section, highlighting possible causes.

5.3 System Metrics

The training time for all experiments is displayed in Table 5. LoRA models took more than twice the time of the fully fine-tuned models. Although

Table 4: Quadratic Weighted Kappa scores of experiments across essay sets and compared to baselines, together with averages (μ). The best overall scores are bolded. The best model results from this study are in italics.

Experiment Configuration	**Essay Set 1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	μ
Full, Uninstructed, Task-Specific	0.82	*0.76*	0.68	*0.69*	0.62	0.75	*0.70*	0.62	0.80	0.66	0.71
Full, Uninstructed, Combined	0.78	*0.76*	0.65	*0.69*	0.78	0.80	*0.70*	0.66	0.80	0.74	*0.74*
Full, Instructed, Task-Specific	*0.84*	0.74	0.69	0.67	*0.81*	*0.85*	0.66	0.59	0.80	0.73	*0.74*
Full, Instructed, Combined	0.77	0.74	*0.70*	0.68	0.77	0.81	0.65	**0.67**	*0.81*	*0.77*	*0.74*
LoRA, Uninstructed, Task-Specific	0.02	0.06	0.07	0.00	0.18	0.02	0.00	0.03	0.41	0.06	0.08
LoRA, Uninstructed, Combined	0.11	0.10	0.31	0.12	0.13	0.16	0.05	0.17	0.38	0.27	0.18
LoRA, Instructed, Task-Specific	0.00	0.03	0.02	-0.01	0.23	0.08	0.07	0.15	-0.07	0.00	0.05
LoRA, Instructed, Combined	0.05	0.03	0.22	0.09	0.28	0.26	0.16	0.08	0.14	0.04	0.13
Uninstructed Baseline BERT	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Instructed Baseline BERT	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00
Kumar et. al. (2020)	**0.87**	**0.82**	**0.75**	**0.74**	**0.85**	**0.86**	**0.73**	0.62	**0.84**	**0.83**	**0.79**

LoRA is considered computationally less intensive and each epoch trains faster, it requires significantly more epochs to reach the early stopping criterion, which full fine-tuning meets way faster. At each update, LoRA updates only a limited number of updates, thereby, the variance of the data is captured more gradually, whereas the full fine-tuning approach manages to converge in only a few epochs[1].

5.4 Discussion

The models of this study achieve comparable or in one case superior (essay task 8) performance, compared to the state-of-the-art machine learning model of Kumar et al. (2020). A significant advantage of this research is that no prior feature engineering was done, which is an integral process of traditional machine learning systems deployed in the field of ASAG.

The action of hand-engineering variables to automate short-answer responses involves selecting lexical, syntactic, and grammatical features that encapsulate the interpretation and evaluation of students' answers. Nonetheless, this process fails to take context into consideration, is prone to human bias, and has limited flexibility across domains. For instance, the random forest-based model of Kumar et al. (2020) extensively defines lexical diversity, content overlap, and vector embeddings as an attempt to effectively capture relevant aspects of student responses. While it is a state-of-the-art automated grading model, it is constrained by domain assumptions.

[1]Epochs are not included in the comparative analysis due to the large number of models. For full training statistics, see the Jupyter Notebook in the repository.

In contrast, the BERT-based models in this paper do not rely on feature engineering but rather are fine-tuned on the essay sets without modifications (uninstructed experiments), or the inputs are slightly altered as a way of instruction-engineering the models, which is favorable in certain tasks, as highlighted previously. This leverages the use of a pre-trained foundational model, which allows the methods in this study to capture various and contextualized representations without manually defined features or explicit instructions regarding the underlying grading criteria.

Table 5: Runtime of all experiments (NVIDIA V100 GPU, 16GB of RAM, 8 CPU cores).

Experiment Configuration	**Runtime (hh:mm:ss)**
Full, Uninstructed, Task-Specific	00:23:30
Full, Uninstructed, Combined	00:24:54
Full, Instructed, Task-Specific	00:28:24
Full, Instructed, Combined	00:24:56
LoRA, Uninstructed, Task-Specific	01:13:01
LoRA, Uninstructed, Combined	01:11:50
LoRA, Instructed, Task-Specific	01:19:34
LoRA, Instructed, Combined	01:00:46

We believe the LLM approach in the field of ASAG is highly beneficial for that specific reason. In an educational setting, students' responses can vary widely across and within tasks, depending on previous education, level of preparation, and overall background. The ability of BERT-based models to capture those variances is a substantial advantage as it allows for a scalable solution across domains without customization for each specific task.

Going deeper in this direction, conventional machine learning approaches often rely on task-specific models and systems, requiring adjustments when used in a different setting. This need for specificity arises from the overfitting problem in machine learning and even earlier neural models like LSTMs (Saha et al., 2019) - generalization across tasks without risking performance remains a challenging objective of those methods. The use of a foundational model eliminates the need for expected response structure or recognition of specific features.

The combined BERT models in the experiments show strong and promising generalizability across the essay tasks, which were in the domain of English literature, Science (Chemistry, Biology, Physics), and Art. The combined models effectively discern relevant aspects of the grading task and achieve nearly identical performance to the task-specific configuration. This suggests that the embeddings in the combined BERT model are robust enough to capture task-specific grading criteria without compromising performance. In educational settings, this aspect is particularly valuable whenever there is a demand for a single, scalable, and efficient solution.

For instance, in the sphere of online courses and assessments, educators could utilize the generalizability of an LLM to be responsible for the participants' evaluation during the course on several assignments, varying in grading scales, descriptions, and expectations, without customizing it for every single one. Moreover, deploying such a system would result in significant cost and labor savings. Another strong point of LLMs is the ability to generate text, which could be used to also provide comprehensive and constructive feedback to students. This is discussed in the Limitations and Future Directions section.

Another important consideration is the impact of instruction-prompting observed in the study. While task-specific fully fine-tuned models benefited from the addition of an instruction in the prompt (sets 5, 6, 10), its overall role remains vague. In particular, the inclusion of instruction in the inputs, when applied to the combined fully fine-tuned models, resulted in a barely noticeable effect. This suggests that the combined models already generalize enough so that the instruction can be discarded in the specific case. This could be somewhat expected, considering the simplicity of the instruction used in this study. Nonetheless, instruction-tuning remains an effective approach across tasks (Zhang et al., 2024), and a suggested improvement is presented in the Future Directions sections, emphasizing the use of the task description in the fine-tuning process.

Lastly, the LoRA fine-tuning approach showed significant limitations when applied to the ASAG task. Even though this method is designed for computational efficiency and reduced memory usage, it took longer to execute, compared to full fine-tuning. The underlying cause of this was that even though each epoch took a shorter amount of time, the overall number of epochs before the early-stopping mechanism was significantly larger, as LoRA had a more gradual drop in validation loss per epoch (see Jupyter Notebook in the repository for full experimental information). Moreover, it introduced several hyperparameters, which have to be further tuned for best performance. Practically, this also makes the LoRA approach harder to implement in a real setting, where simplicity of solutions would be preferred.

Additional hyperparameter tuning of a LoRA model would require a sophisticated search (e.g. grid, random, or Bayesian) to identify optimal configurations, necessitating additional resources. As discussed previously, educational environments require models that perform consistently across various domains of tasks with minimal or no reconfiguration. Lastly, its underperforming results could be further explained by the fact that low-rank adaptations are less capable of capturing detailed, nuanced, and context-sensitive patterns, common in ASAG. This makes the LoRA approach unsuitable for scalable and efficient grading systems with minimal tuning, as required in the automated grading field.

6 CONCLUSION

6.1 Summary of Contributions

While many works on automated short-answer grading rely on hand-crafted feature engineering to capture superficial linguistic features, more recent applications of large language models often struggle to generalize across diverse tasks. Our work advances the ongoing discussion by implementing a fine-tuned LLM approach tailored specifically to the task of short-answer grading. By leveraging the deep contextual understanding of LLMs, our model captures nuanced aspects of meaning and coherence in student responses, going beyond the limitations of surface-level feature extraction. This reduces the reliance on human-engineered features and mitigates the biases they can introduce, allowing the model to learn directly from data. The experimental results in this study showed that fine-tuned models are able to achieve high agreement with human graders, delivering robust scoring outcomes closely aligned with expert evaluations. Furthermore, this study considers a combined model experimental strategy, enabling the model to handle a variety of tasks with differing grading rubrics and scales without the need for separate fine-tuning, enhancing the adaptability and scalability of the grading process.

6.2 Limitations and Future Work

The impact of instruction-prompting on model performance remains unclear. While some essay sets showed improvements, others experienced performance degradation. Current prompting strategies include only brief task descriptions, excluding the specific question answered by the student. This limitation hinders the model's ability to differentiate well-written but irrelevant answers from those that are accurate and relevant. Future work could expand input contexts to include task descriptions, but this will require models with longer context windows, increasing computational demands. Research into memory-efficient methods for managing extended contexts while preserving accuracy could address this challenge. The lack of a rigorous evaluation of generalization remains a concern. The data split used in this study shuffled samples across training, validation, and test sets, precluding an unbiased assessment of model performance on entirely unseen tasks. Future studies could evaluate models on new, unseen essay tasks to better understand their zero-shot capabilities. This would provide insights into their adaptability and support the development of ASAG systems robust enough for deployment across diverse educational contexts. Additionally, exploring continual learning approaches could ensure sustained performance on previously learned tasks while adapting to new ones.

Moreover, current models only provide a score, limiting their educational utility. Effective feedback is crucial for student learning, and future work could leverage LLMs' text-generation capabilities to provide personalized, actionable feedback alongside grades. However, this approach must be rigorously validated to ensure accuracy and reliability. Poorly designed feedback risks being misleading or overly simplistic, potentially harming learning outcomes. Conversely, responsibly implemented feedback mechanisms could transform automated grading systems into comprehensive educational tools that serve as both assessors and tutors. The deployment of LLMs in educational settings raises critical ethical concerns. These systems influence students' academic and professional trajectories, necessitating measures to address biases, quantify uncertainties, and ensure transparency in training data and protocols. Future research should prioritize the development of bias assessment techniques to ensure fair outcomes across diverse student populations, fostering trust in the system's reliability and equity.

Key strategies for implementing explainability and interpretability include posthoc explainability methods like SHAP and Integrated Gradients, as explored in Tornqvist et al. (2023) to connect model predictions with interpretable token- or sentence-level justifications. Additionally, Aggarwal et al. (2024) introduce the EngSAF dataset, containing questions and responses from multiple engineering domains, and use it along the Label-Aware Synthetic Feedback Generation (LASFG) strategy to provide detailed, content-focused synthetic feedback alongside grades. Further research should be aimed at assessing various strategies for feedback generation in the interest of equipping teachers with a valuable tool for providing systematic feedback and improving learning efficiency.

ACKNOWLEDGEMENTS

We thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Hábrók high-performance computing cluster, using an NVIDIA V100 GPU, 16GB of RAM, and 8 CPU cores. OpenAI's ChatGPT was used to enhance the readability of text.

REFERENCES

Aggarwal, D., Bhattacharyya, P., and Raman, B. (2024). ”i understand why i got this grade”: Automatic short answer grading with feedback.

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

Barbara, Ben Hamner, J. M., lynnvandev, and Shermis, M. (2012). The hewlett foundation: Short answer scoring.

Biderman, D., Portes, J., Ortiz, J. J. G., Paul, M., Greengard, P., Jennings, C., King, D., Havens, S., Chiley, V., Frankle, J., Blakeney, C., and Cunningham, J. P. (2024). Lora learns less and forgets less.

Bonner, E., Lege, R., and Frazier, E. (2023). Large language model-based artificial intelligence in the language classroom: Practical ideas for teaching. *Teaching English with Technology*, 23(1):23–41.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Ding, Q., Ding, D., Wang, Y., Guan, C., and Ding, B. (2023). Unraveling the landscape of large language models: a systematic review and future perspectives. *Journal of Electronic Business & Digital Economics*.

Doewes, A., Kurdhi, N., and Saxena, A. (2023). Evaluating quadratic weighted kappa as the standard performance metric for automated essay scoring. In *Proceedings of the 16th International Conference on Educational Data Mining*, pages 103–113. International Educational Data Mining Society (IEDMS).

Galhardi, L. and Brancher, J. (2018). Machine learning approach for automatic short answer grading: A systematic review. pages 380–391.

Godbole, V., Dahl, G. E., Gilmer, J., Shallue, C. J., and Nado, Z. (2023). Deep learning tuning playbook. Version 1.0.

Haller, S., Aldea, A., Seifert, C., and Strisciuglio, N. (2022). Survey on automated short answer grading with deep learning: from word embeddings to transformers.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). Lora: Low-rank adaptation of large language models.

Kortemeyer, G. (2023). Performance of the pre-trained large language model gpt-4 on automated short answer grading.

Kumar, Y., Aggarwal, S., Mahata, D., Shah, R. R., Kumaraguru, P., and Zimmermann, R. (2020). Get it scored using autosas – an automated system for scoring short answers.

Latif, E. and Zhai, X. (2024). Fine-tuning chatgpt for automatic scoring. *Computers and Education: Artificial Intelligence*, 6:100210.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization.

Mohler, M., Bunescu, R., and Mihalcea, R. (2011). Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In Lin, D., Matsumoto, Y., and Mihalcea, R., editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 752–762, Portland, Oregon, USA. Association for Computational Linguistics.

Page, E. B. (1966). The imminence of... grading essays by computer. *The Phi Delta Kappan*, 47(5):238–243.

Reizinger, P., Ujváry, S., Mészáros, A., Kerekes, A., Brendel, W., and Huszár, F. (2024). Position: Understanding llms requires more than statistical generalization. In *Forty-first International Conference on Machine Learning*.

Saha, S., Dhamecha, T. I., Marvaniya, S., Foltz, P., Sindhgatta, R., and Sengupta, B. (2019). Joint multi-domain learning for automatic short answer grading. *arXiv preprint arXiv:1902.09183*.

Shallue, C. J., Lee, J., Antognini, J. M., Sohl-Dickstein, J., Frostig, R., and Dahl, G. E. (2018). Measuring the effects of data parallelism on neural network training. *CoRR*, abs/1811.03600.

Sung, C., Dhamecha, T. I., and Mukhi, N. (2019). Improving short answer grading using transformer-based pre-training. In Isotani, S., Millán, E., Ogan, A., Hastings, P., McLaren, B., and Luckin, R., editors, *Artificial Intelligence in Education*, pages 469–481, Cham. Springer International Publishing.

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

Tashu, T. M., Maurya, C. K., and Horvath, T. (2022). Deep learning architecture for automatic essay scoring.

Tornqvist, M., Mahamud, M., Mendez Guzman, E., and Farazouli, A. (2023). ExASAG: Explainable framework for automatic short answer grading. In Kochmar, E., Burstein, J., Horbach, A., Laarmann-Quante, R., Madnani, N., Tack, A., Yaneva, V., Yuan, Z., and Zesch, T., editors, *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 361–371, Toronto, Canada. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention is all you need.

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. (2021). Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Zhang, A., Lipton, Z. C., Li, M., and Smola, A. J. (2023a). *Dive into Deep Learning*, pages 551–552. Cambridge University Press. https://D2L.ai.

Zhang, S., Dong, L., Li, X., Zhang, S., Sun, X., Wang, S., Li, J., Hu, R., Zhang, T., Wu, F., and Wang, G. (2024). Instruction tuning for large language models: A survey.

Zhang, Y., Cui, L., Cai, D., Huang, X., Fang, T., and Bi, W. (2023b). Multi-task instruction tuning of llama for specific scenarios: A preliminary study on writing assistance. *arXiv preprint arXiv:2305.13225*.

Zhao, Z., Fan, W., Li, J., Liu, Y., Mei, X., Wang, Y., Wen, Z., Wang, F., Zhao, X., Tang, J., et al. (2023). Recommender systems in the era of large language models (llms). *arXiv preprint arXiv:2307.02046*.