

Generative Language Models for Argumentation Annotation

Kai North¹, Christopher Ormerod²

¹Cambium Assessment ²College Board

Correspondence: Kai.North@cambiumassessment.com

Abstract

This study explores the integration of a generative language model (GLM) into an Automated Writing Evaluation (AWE) system designed to highlight both argumentative components and errors in spelling and grammar. We fine-tune an open-source GLM using parameter-efficient techniques. We evaluate the model's capabilities in both argument analysis and error detection against established datasets. We demonstrate that a single GLM with parameter-efficient adapters can accurately identify argumentative clauses, classify their types, map relationships between them, and flag mechanical mistakes. We establish that our AWE system performs at human-level accuracy, while only requiring a fraction of the computational power of much larger models.

1 Introduction

Over the past two decades, Automated Writing Evaluation (AWE) systems have proven themselves to be a beneficial technology in educational assessment, offering promising solutions to longstanding challenges in the field of education (Huawei and Aryadoust, 2023). Generative language models (GLMs) have demonstrated remarkable abilities across an increasingly diverse range of tasks, prompting researchers and educators alike to explore how these capabilities can be effectively applied to benefit both teachers and students in educational contexts (Gimpel et al., 2023).

This article focuses specifically on demonstrating a GLM's capabilities in two areas: computational argumentation, a specialized field within artificial intelligence (AI) that concentrates on the systematic analysis of arguments (Eger et al., 2017), and grammatical error identification (Yanakoudakis et al., 2011). Computational argumentation has several valuable practical applications in both legal and educational contexts where argument quality is essential. Given that GLMs have

demonstrated the capacity to comprehend and synthesize complex arguments, we show that they are especially well-suited for computational argumentation applications. Secondly, since the models are pretrained on a large corpus of predominantly well-formatted text, it should be relatively straightforward to determine when a piece of text deviates from conventional norms of language, which makes it ideal for detecting spelling and grammatical errors.

Using the Argument Annotated Essays (AAE) corpus (Stab and Gurevych, 2014, 2017) and the error annotated Cambridge Learning Corpus (CLC), we demonstrate that fine-tuned Low-Rank adapters (Hu et al., 2021) for the Llama 3 model (AI@Meta, 2024) can accurately perform four distinct AWE-centered tasks:

- Identifying argumentative clauses within student essays.
- Classifying these clauses as either Major Claims, Claims, or Premises.
- Determining whether clauses are supportive, opposing, or unrelated.
- Identifying errors in spelling and grammar.

We start by reviewing some of the existing literature in §2 followed by a description of the dataset used in this study §4. This includes how it is defined in terms of the tasks of annotating and classifying components and stance identification. We then describe the methods in §5 covering the models we use, the type of prompt-templates they require, and the fine-tuning methods applied to arrive at the final results in §6. §7 lastly provides an overview and discussion.

2 Background and Related Research

AWE systems have naturally evolved in parallel with Automated Essay Scoring (AES) systems.

In addition to spelling and grammatical features, AWE systems incorporate rule-based features developed for AES systems to provide feedback to students (Huawei and Aryadoust, 2023). These features include syntactic complexity measures such as sentence length, variation and structural diversity, lexical sophistication indicators such as vocabulary richness and word choice appropriateness, coherence metrics that evaluate textual organization and argument flow, and content relevance measures that assess topic adherence (Attali, 2013). A criticism of this approach is that both its features and feedback ignore the essay's semantic content. Moreover, because scoring and feedback rely on the same features, any score improvement may reflect the increased presence of a set of features rather than a student's genuine adherence to the rubric.

In the search for more specific evaluation based on the semantic content of the essay, it makes sense to consider the integration of GLMs into the essay evaluation process. Mizumoto and Eguchi (2023) explored how well these models can be prompted to score the ETS Corpus of Non-Native Written English (TOEFL11). The work of Xiao et al. (2024) used prompts with GPT-4 and a fine-tuned version of GPT-3.5 with the Kaggle ASAP AES dataset (Shermis and Hamner, 2013) to achieve agreement levels comparable, but inferior to extended versions of BERT (Yang et al., 2020; Uto and Uchida, 2020). Other researchers have been successful using large closed-source models and chain-of-thought prompting to improve correlation with human-assigned text quality ratings (Chiang and Lee, 2023; Liu et al., 2023). Another approach is to use an LLM to derive supervised features, which can then be used to provide accurate scoring or feedback (Hou et al., 2025). This approach is very closely related to trait-based scoring, which is a way of scoring that assigns scores to each of the different aspects of writing (Mathias and Bhattacharyya, 2018).

Due to the computational demands of GLMs, many researchers rely on the use of an Application Programming Interface (API), which poses significant problems in privacy and security when applied to large-scale assessment (Bulut et al., 2024). An approach to scoring that circumvents these concerns uses parameter efficient fine-tuning (Xu et al., 2023). In particular, researchers showed that the combination of quantization and Low-Rank Adapters (QLoRA) (Dettmers et al., 2023), applied with open-source models to the Kaggle dataset

provided agreements that were higher than those achieved by much larger models (Ormerod and Kwako, 2024). In addition to higher agreements, these models use scores as anchors for some limited feedback, however, this feedback still requires careful validation. At this point, much of the validation for GLM outputs are dependent on subject matter expert (SME) review. While SME review is a valuable process, it is also expensive to implement.

Automated validation is particularly effective for two types of writing feedback: grammatical correctness and argument structure. Assessing grammatical quality involves comparing a given text against the output of models specifically fine-tuned for spelling and grammatical error correction (Rothe et al., 2021). The performance of these automated systems can be benchmarked using established datasets such as JFLEG for fluency edits (Napoles et al., 2017) and the CLC FCE dataset for learner English (Yannakoudakis et al., 2011). Historically, before the widespread adoption of modern Generative Language Models (GLMs), significant effort was dedicated to extending earlier architectures like the T5 model to handle more complex stylistic and coherence-based edits (Raheja et al., 2023). Today, however, such corrective and stylistic feedback can often be obtained more directly by prompting large-scale GLMs.

The other task of interest is the identification and classification of argumentative clauses. There are two datasets relevant to this task; the PERSUADE Corpus (Crossley et al., 2022) and the AAE corpus (Stab and Gurevych, 2014, 2017). It has been shown that traditional mask-language models, such as BERT (Devlin et al., 2018) and XLNet (Yang et al., 2019), can be effectively fine-tuned for argumentation annotation (Habernal et al., 2023). When using traditional LLMs, this task uses a collection of models; a token classifier to determine and classify argumentative clauses which can be used with or without an IOB-tagger, while another model is used to determine the relations between stances (Ormerod et al., 2023). When we approach this task using GLMs, we can essentially use the same techniques applied to scoring established in (Ormerod and Kwako, 2024) to produce free-form text that adheres to a particular template that facilitates simple parsing.

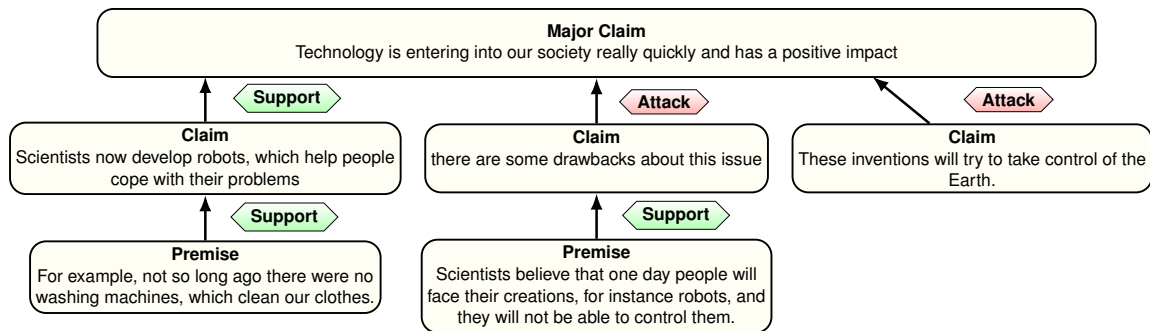


Figure 1: An example of the supporting and attacking structures of an argument within an essay.

3 Study Purpose

This article highlights performance in two key areas: computational argumentation, and grammatical error identification by exploring the following two research questions.

- To what extent can a parameter-efficient fine-tuned GLM accurately identify and classify argumentative components and stance relations in student essays compared to established benchmarks?
- How effectively can the same architecture detect and label spelling and grammatical errors using standardized evaluation datasets?

This articles therefore provides the performance of a fine-tuned GLM compared to traditional approaches for these tasks. This article also provides insight into how these models may be deployed within AWE systems.

4 Data

The AAE corpus consists of 402 essays taken from essayforum.com. The forum is an active online community that provides feedback and suggestions for a range of essays. The essays selected from the website were chosen at random. The chosen essays are written by students of a variety of ability levels and are on a diverse set of topics. These essays were expertly annotated at the clause level by trained annotators. Annotators were provided with a comprehensive annotation scheme as supplied by [Stab and Gurevych \(2014, 2017\)](#). Annotators were made familiar with this annotation scheme over the period of several training sessions whereby annotators worked together to annotate example essays and discuss disagreements. The 402 essays annotated after annotators' training saw an inter-rater

agreement of 0.7726, 0.6033, and 0.7594 Krippendorff's alpha for major claim, claim, and premise annotation, respectively.

Component Identification Annotators assigned word-level BIO tags to identify the start and end of every component and separate argumentative from non-argumentative text in each essay. Annotators assigned "**B**" to mark the first word of each component, "**I**" to each of their consecutive words, and "**O**" to those words which existed outside of, and were not part of, any identified component.

Component Classification Alongside marking component boundaries, annotators were also asked to consider whether each component was characteristic of either a "*major claim*", "*claim*", or "*premise*". These were defined as follows:

- **Major Claim** - Outlines the author's stand-point on a given topic. Is considered the primary position or argument that the essay is making. The first component identified in Figure 1 is an example of a major claim: "*technology is entering into our society really quickly*". It is the main argument provided by the author which is supported by the following components.
- **Claim** - Claims are the building blocks of an overall argument. They provide reasoning that supports the major claim. The second component in Figure 1 was labeled as a claim: "*Scientists now develop robots, which help people cope with their problems*". This component provides additional reasoning that helps to aid the major claim by illustrating the speed of technology advancement.
- **Premise** - A plausible fact that serves as the basis of a claim. The last component recognized in Figure 1 was marked as a premise:

Task	Label	Train	Test
Identification	B	4,820	1,265
	I	71,687	17,797
	O	28,841	7,105
Classification	Major Claim	598	153
	Claim	1202	304
	Premise	3023	809
Stance	Support	2846	767
	Attack	177	42

Table 1: A summary of the number of elements to classify.

“For example, not so long ago there were not washing machines, which clean your clothes”. It provides a fact in support of the aforementioned major claim and claim.

Stance Identification Using the AAE corpus, we are also able to identify the relationship between each major claim, claim, and premise. Annotators were tasked with identifying whether a given claim or premise (source) either “*supports*” or “*attacks*” another claim or major claim (target), or whether the two clauses are unrelated. Source and target pairs that showed the source as having agreement with or evidence in favor of the target were labeled as “*support*”. In contrast, pairs that showed the source as having disagreement or evidence against the target were marked as “*attack*”. This is illustrated in Figure 1.

In summary, this dataset consists of 402 essays, with a total of 147721 word-level BIO-tags accounting for 6089 different argumentative components, with 3832 relations. There is a well-defined test-train split at the essay level dividing the total number of tokens tags, argumentative components, and relations into a train-test split for each task. A summary of the number of tags and classifications is given in Table 1.

Error Detection The Cambridge Learner Corpus (CLC; Berzak et al., 2024) was used for investigating the performance of our fine-tuned GLM at spelling and grammatical error detection. The CLC contains scripts from the First Certificate in English (FCE) exam. The FCE is designed for upper-intermediate English language learners. The dataset comprises of free-text responses to writing prompts, including letters, reports, articles, compositions, or short stories, with each script being 200-400 words long. Each response has an overall grammatical and spelling correctness score. The

Split	Sentences	Correct (C)	Incorrect (I)	% Incorrect
Train	1558	27281	1893	6.94%
Dev	222	4099	286	6.98%
Test	446	7951	555	6.98%

Table 2: Summary of Token Counts in Cleaned FCE Dataset.

CLC was chosen due to its provision of this score.

A derived version of the dataset (FCE-error-detection) also exists with each individual token being assigned a correct (c) or incorrect (i) label depending on whether it contained a spelling or grammatical mistake or preceded a missing word or phrase. We further cleaned this version of the dataset by removing all incorrect (i) labels that were assigned to words preceding a missing word or phrase or that were wrongly assigned as incorrect in the original dataset. These included acronyms, such as “U.S.A.”, proper nouns, for instance “*Etruscans*”, and abbreviations: “*1st*”, “*2nd*”, “*3rd*” and so on. The remaining spelling and grammatical errors were used to train our AWE system for error detection and were presented to our model in their original contexts (sentences) amounting to the following new train, dev, and test splits shown in Table 2.

The JHU Fluency-Extended GUG corpus (JF-LEG; Napoles et al., 2017) was also used to determine the performance of our AWE system at correcting identified errors. This corpus consists of 1,502 sentences produced by English language learners that contain grammatical or spelling errors. These sentences were corrected by four Mechanical Turk workers. Four corrected versions were subsequently produced for each original sentence. The original sentences were used as input to our AWE models with their corresponding corrections being used to evaluate our model’s performance. 755 sentences were used from training and development with the remaining 747 sentences being used for evaluation.

4.1 Prompt Templates

Prompt templates were constructed for each of the above tasks (Summarized within Table 3; for a full list of prompts see Appendix). These prompt templates were designed to mimic the thought-process of the human annotator in an attempt to minimize the variance between LLM predictions and human assignments. The inputted prompt template for each task is provided and described below. A de-

Task	Prompt Summary
Component Identification	User: List argumentative clauses from essay. Assistant: Returns list of clauses.
Component Classification	User: Label component as <i>major claim</i> / <i>claim</i> / <i>premise</i> using rubric. Assistant: Outputs class label.
Stance Identification	User: Given statement + claim, classify as <i>supports</i> or <i>attacks</i> . Assistant: Outputs relation.
Error Detection (FCE)	System: Spellchecker. User: List misspelled words only (bulleted). Assistant: Outputs incorrect words.
Error Correction (JF-LEG)	System: Spellchecker. User: Correct full sentence. Assistant: Outputs corrected sentence.

Table 3: Overview of prompts used across tasks.

scription of our chosen LLM and training strategy is provided with Section 5.

Component Identification Prompt Our component identification prompt simply asks for any argumentative clauses within a given essay. This is because the original annotators were asked to identify any argumentative components at a clause level (Stab and Gurevych, 2014, 2017). When the model was prompted to identify these clauses, the output was presented with a restatement of the task followed by an enumerated list. We can parse the output by identifying those elements of the list using regular expressions. This means that we have a list of true clauses identified by the human annotator, and a list of predicted clauses, predicted by the model.

We convert the annotations into a set of BIO-tags by matching the longest sequence of tokens identified by both human annotators and the GLM across the tokenized essay. It is worth stating that the outputs of the GLM capitalized the first word in the clause, hence, we used case-insensitive token-level matching to define true and predicted BIO-tags for each token. When training the GLM to match the output, we formatted the list so that the clauses appear in the same order in the essay itself.

Component Classification Prompt The prompt designed for component classification took as input the entire essay, all identified components, as well as the official rubric provided to the human annotators. The provided rubric defined the characteristics of a major claim, claim, and premise. The LLM was tasked with using the provided rubric to label each of the listed components. This was done for all components present within the entire essay,

since the prediction of one label, such as a major claim, inherently influences the prediction of another: claim or premise. We highlighted the major claim, claim, and premise in the prompt by using the markdown for bold (i.e., **text**). When fine-tuning the model, this meant that we could identify the class of the component by extracting the element in bold in the output.

Stance Identification Prompt The stance identification prompt asked the LLM to evaluate the relationship between two clauses: a claim and a related statement. Rather than using broad concepts of support or attack, the LLM was specifically asked to interpret support to mean that both the claim and statement could be simultaneously true. This more precise logical interpretation produced better agreement scores compared to more general support/attack relationship. Without this specification, it seemed to interpret the term support in argumentation as equivalence, which is far stronger than the intended interpretation. We also highlighted supports and attacks in bold in the prompt so that we could extract the bold text in the LLM output to determine the classification.

Error Detection Prompt Multiple rounds of experimentation were conducted to uncover the most effective prompt for error detection. Using the FCE dataset, we provide the LLM with a pre-tokenized list of all the words in the inputted sentence (arg1). This was done so as to better align the LLM’s output with the original sentence reducing the likelihood of unwanted text being generated. The LLM is then tasked with identifying which of these words contain a spelling or grammatical error and to produce its answer in the format of a bulleted list.

An additional prompt was constructed to determine our model’s performance for identifying errors and producing valid corrections. Sentences with grammatical and spelling mistakes were taken from the JFLEG corpus (Napoles et al., 2017) and inputted into the LLM using the prompt shown in Figure 6 (Appendix). The prompt provides the original sentence and tasks the LLM to correct any spelling and grammatical mistakes. The LLM then produces only a corrected version of the inputted sentence which is compared against four gold standard sentences also taken from the JFLEG corpus.

5 Method

5.1 Model Training

The model we employ is based on a modified Llama 3.1 architecture (Touvron et al., 2023). This particular variant has been fine-tuned using instruction-based training and contains approximately 8 billion parameters. The model utilizes a decoder-only transformer structure with 32 layers. Each attention layer has a dimension of 4,096, complemented by a feed-forward layer of 14,336 dimensions. The model’s vocabulary encompasses 128,000 tokens, and it can natively process input sequences of up to 8,192 tokens in length.

The availability of the model weights allows us to run the model locally. This is particularly important in educational settings where the security of student data is an important factor (Bulut et al., 2024). To maximize the accuracy of these models on the specific task on our modest hardware, we are required to explore parameter-efficient methods (Xu et al., 2023). In particular, we seek to fine-tune this model using the Quantized Low-Rank adapters (QLoRa) method demonstrated by Dettmers et al. (Dettmers et al., 2023).

5.2 Fine-tuning with QLoRA

QLoRA is a powerful and parameter-efficient method of fine-tuning. It was recently used to demonstrate the ability of GLMs to perform AES and provide feedback (Ormerod and Kwako, 2024).

At the core of the Llama architecture (Touvron et al., 2023), and transformers more generally, is the attention mechanism (Vaswani et al., 2017). The attention mechanism relies on multiple linear layers, taking the canonical form

$$L(x) = Wx + b. \quad (1)$$

The LoRa method replaces (1) with the operator

$$\tilde{L}(x) = (W + \Delta)x, \Delta = BA \quad (2)$$

If $W \in \text{Mat}_{m \times n}$ then $B \in \text{Mat}_{m \times k}$ and $A \in \text{Mat}_{k \times n}$ for some k . The multiplication, $\Delta = BA$ is a convenient parameterization of an element of $\text{Mat}_{m \times n}$ of a rank that is less than or equal to k , which is assumed to be less than m , hence the term Low-Rank adaptation. In this method, the matrix W is not trainable; we are only applying updates to the matrices A and B using the memory efficient 8-bit paged weighted Adam optimizer (Dettmers et al., 2022). While LoRa is a general technique

applied to any of the linear layers of a model, it is commonly applied to the linear layer applied to obtain the keys, queries, and values.

The instruction tuning phase of the model training for the Llama models includes a large number of diverse tasks, in a similar manner to the T5 model (Raffel et al., 2020). More generally, GLMs that have been instruction-tuned seem to exhibit a degree of transfer learning. It stands to reason that the three respective tasks associated with the AAE corpus may benefit from transfer learning. This means that, instead of training a model for each task, transfer learning facilitates the use of a single model for all three tasks. We do this simply by combining prompts associated with component identification, component classification, and stance identification into one dataset. Unlike our previous experiments evaluating our model on each task independently, for transfer learning our training corpus consists of prompts for all tasks in combination.

5.3 Validation and Metrics

F1-score was used to evaluate model performance for each task. The predictions made for component identification were used to create a BIO tagged vector. This resulted in a vector containing three class predictions of “*B*”, “*I*”, and “*O*” which were compared to those provided by the human annotators. The multi-class and binary labels of the component classification, stance identification and error detections tasks were also compared to those produced by the human annotators. F1-score was calculated as follows:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

where

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP + FP} \\ \text{Recall} &= \frac{TP}{TP + FN} \end{aligned}$$

where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. For multiple classes, we will be reporting macro-averages over the classes, which are unweighted averages of the F1-scores over each class.

GLEU score was used to evaluate our LLM’s output for producing full sentence corrections using the JFLEG corpus. GLEU score is based on

Model	#	Learning Strategy	Identification				Classification				Stance
			F1	F1 B	F1 I	F1 O	F1	F1 MC	F1 CI	F1 Pr	F1
Llama-3-8B	-	Base	0.396	0.176	0.572	0.410	0.380	0.803	0.428	0.287	0.624
	-	Task-Specific Fine-tuning	0.920	0.888	0.958	0.915	0.891	0.937	0.803	0.933	0.887
	-	Multitasking Fine-tuning	0.922	0.906	0.957	0.911	0.889	0.945	0.789	0.932	0.845
Baselines	-	Human	0.886	0.821	0.941	0.892	0.868	0.926	0.754	0.924	0.844
	-	Results of (Stab and Gurevych, 2017)	0.867	0.809	0.934	0.857	0.826	0.891	0.682	0.903	0.702
	-	XLNet Base	0.936	0.956	0.951	0.947	0.847	0.924	0.814	0.950	0.827
	-	XLNet Large	0.933	0.954	0.951	0.946	0.870	0.932	0.846	0.940	0.885
	-	Claude 3.5 Sonnet	0.460	0.385	0.536	0.459	0.820	0.841	0.690	0.897	0.645

Table 4: The results of Llama 3-8B-Instruct and fine-tuned derivatives against several benchmarks.

Model	Learning Strategy	FCE			JFLEG
		F1	F1 Correct (c)	F1 Incorrect (i)	GLEU
Llama-3-8B	Base	0.834	0.973	0.694	0.542
	Fine-tuned	0.843	0.975	0.710	0.552

Table 5: The results of Llama 3-8B-Instruct and fine-tuned derivative for error detection.

the machine translation metric BLEU score and is designed specifically for the task of error correction (Napoles et al., 2017). GLEU score represents the n-gram overlap between the model’s output and human corrected sentences. It also penalizes n-grams that were changed in the human corrected sentences but left unchanged by a model.

6 Results

In addition to our own results, for component identification, classification and stance detection we compare our results to several key benchmarks. These include the benchmark set by human annotators, reported in the original paper alongside a frequency and rule-based approach (Stab and Gurevych, 2017), the recent results using XLNet (Ormerod et al., 2023), and the application of a version of Anthropic’s Claude model and parsing methods to the prompts used. Table 4 depicts model performance per task alongside other benchmarks.

We readily see that the Llama model performs well against available benchmarks. In fact, the model performs about as well as human annotators, however, the model did not perform as well as the combination of a collection of XLNet models fine-tuned for each task. While the Claude model was one of the worst-performing models, it is somewhat surprising that the Claude model performed so well in the classification task with no fine-tuning.

Performances for our error detection model on both the FCE and JFLEG datasets can be seen in Table 5. The identification of individual spelling and grammatical mistakes was evaluated using F1-score, whereas the performance of generating correct sentences using the JFLEG dataset was

reported using GLEU. Llama achieved a macro-average F1-score of 0.843 for error detection with individual class performances being an F1-score of 0.975 for identifying correct words and 0.710 for recognizing incorrect words. Unsurprisingly, detecting correct words was therefore an exceptionally easier task. Words that were wrongly assigned as being incorrect were often due to Llama’s unfamiliarity with grammatical forms, for instance, believing that the “s” in “*solutions*” was grammatically incorrect, the presence of unfamiliar out-of-vocabulary words, or words that Llama believed did not fit the provided context written in English, such as “*Buenos Aires.*” Additionally, it was discovered that fine-tuning Llama on the FCE dataset over 10 epochs did little to improve its overall performance. It would seem that the task of error detection relies heavily on knowledge acquired during pre-training more so than that obtained from fine-tuning. This suggests that larger models greater than 8B parameters would demonstrate superior performances.

Llama’s performance at generating corrected sentences amounted to a GLEU score of 55.2 on the JFLEG dataset. Human annotators achieved a GLEU score of 55.3 (Napoles et al., 2017). However, other models, including Electra (Liu et al., 2021) and a sequence-to-sequence neural network (Ge et al., 2018) have attained GLEU scores of 61.61 and 62.42, respectively. As such, despite Llama’s ability to generate corrected sentences on par with human annotators, it fails to produce sentences of the same quality as previous approaches. This is due to Llama’s corrected sentences containing additional unnecessary vocabulary rather than simply corrected versions of incorrect words. The sentences produced by Llama were therefore

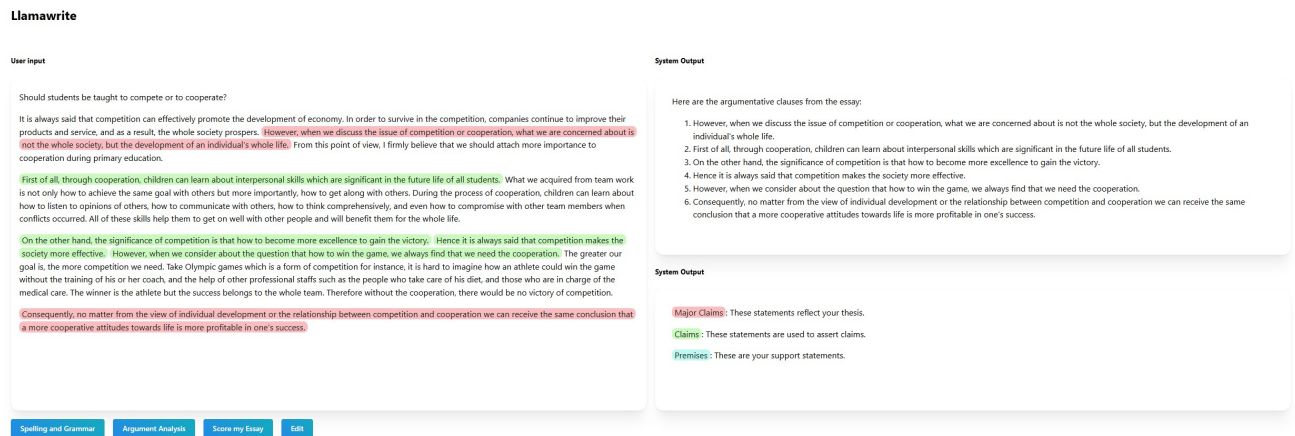


Figure 2: We present a demo of the user interface for our integrated AWE system.

not necessarily incorrect. Instead, they were over-corrected to a degree beyond that required. As a consequence, they were too far removed from the gold standard, hence, negatively affecting Llama’s GLEU score. This can be seen in the below example.

Incorrect: However, I strongly maintained that traveling by their own gives a lot of advantages.

Gold Standard: However, I strongly maintain that traveling by their own has a lot of advantages.

Llama Output: However, I strongly believe that traveling by oneself gives a lot of advantages.

7 Discussion

The results of this study demonstrate significant progress in leveraging a GLM for AWE, while also highlighting important implications for educational technology and writing instruction. Llama was found to perform on par with human annotators for component identification, component classification and stance detection. Llama also performed well at error detection with little change in performance after having been fine-tuned. Our findings therefore suggest that a GLM can be implemented within an AWE with the potential of improving overall performances. We have created a demo of such a system which is presented within Figure 2. As such, the successful implementation of our AWE system demonstrates a departure from traditional AWE approaches. While existing systems have largely relied on predetermined scoring rubrics and

template-based comments, our model shows the capability to identify argumentative elements similar to that of human annotators.

8 Conclusion and Future Work

This research represents a step toward more sophisticated and useful automated writing evaluation systems. By demonstrating the feasibility of generating open-ended, validatable feedback, we have shown new possibilities for supporting writing instruction at scale. Future research should focus on refining these capabilities while carefully considering their implementation in educational contexts. As these technologies continue to evolve, they have the potential to significantly enhance writing instruction and support both teachers and students in the writing process.

9 Limitations

Several limitations warrant attention. First, while our validation framework provides important quality assurance, further research is needed to evaluate the efficacy of GLM-generated feedback on student writing development. Similar to the way in which we accommodated free-form responses in assessment, we need a way to accommodate free-form textual feedback. Additionally, questions remain about how best to integrate such systems into existing educational practices and how to ensure they complement rather than replace human instruction.

References

AI@Meta. 2024. [Llama 3 Model Card](#).

- Yigal Attali. 2013. Validity and Reliability of Automated Essay Scoring. In *Handbook of Automated Essay Evaluation*. Routledge. Num Pages: 18.
- Yevgeni Berzak, Roi Reichart, and Boris Katz. 2024. Cambridge first certificate in english (fce) dataset. Dataset.
- Okan Bulut, Maggie Beiting-Parrish, Jodi M. Casabianca, Sharon C. Slater, Hong Jiao, Dan Song, Christopher M. Ormerod, Deborah Gbemisola Fabiyi, Rodica Ivan, Cole Walsh, Oscar Rios, Joshua Wilson, Seyma N. Yildirim-Erbasli, Tarid Wongvorachan, Joyce Xinle Liu, Bin Tan, and Polina Morilova. 2024. The Rise of Artificial Intelligence in Educational Measurement: Opportunities and Ethical Challenges. *arXiv preprint*. ArXiv:2406.18900.
- Cheng-Han Chiang and Hung-yi Lee. 2023. A Closer Look into Automatic Evaluation Using Large Language Models. *arXiv preprint*. ArXiv:2310.05657 [cs].
- Scott A. Crossley, Perpetual Baffour, Yu Tian, Aigner Picou, Meg Benner, and Ulrich Boser. 2022. The persuasive essays for rating, selecting, and understanding argumentative and discourse elements (PER-SUADE) corpus 1.0. *Assessing Writing*, 54:100667.
- Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2022. 8-bit Optimizers via Block-wise Quantization. *arXiv preprint*. ArXiv:2110.02861 [cs].
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Fine-tuning of Quantized LLMs. *Advances in Neural Information Processing Systems*, 36:10088–10115.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Technical Report arXiv:1810.04805, arXiv. ArXiv:1810.04805 [cs] type: article.
- Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural End-to-End Learning for Computational Argumentation Mining. *arXiv preprint*. ArXiv:1704.06104 [cs].
- Tao Ge, Furu Wei, and Ming Zhou. 2018. Reaching Human-level Performance in Automatic Grammatical Error Correction: An Empirical Study. *arXiv preprint*. Number: arXiv:1807.01270 arXiv:1807.01270 [cs].
- Henner Gimpel, Kristina Hall, Stefan Decker, Torsten Eymann, Luis Lämmermann, Alexander Mädche, Maximilian Röglinger, Caroline Ruiner, Manfred Schoch, Mareike Schoop, Nils Urbach, and Steffen Vandrik. 2023. Unlocking the power of generative AI models and systems such as GPT-4 and ChatGPT for higher education: A guide for students and lecturers. Working Paper 02-2023, Hohenheim Discussion Papers in Business, Economics and Social Sciences.
- Ivan Habernal, Daniel Faber, Nicola Recchia, Sebastian Bretthauer, Iryna Gurevych, Indra Spiecker genannt Döhmann, and Christoph Burchard. 2023. Mining Legal Arguments in Court Decisions. *arXiv preprint*. ArXiv:2208.06178.
- Zhaoyi Joey Hou, Alejandro Ciuba, and Xiang Lorraine Li. 2025. Improve LLM-based Automatic Essay Scoring with Linguistic Features. *arXiv preprint*. ArXiv:2502.09497 [cs].
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint*. ArXiv:2106.09685 [cs].
- Shi Huawei and Vahid Aryadoust. 2023. A systematic review of automated writing evaluation systems. *Education and Information Technologies*, 28(1):771–795.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. *arXiv preprint*. ArXiv:2303.16634 [cs].
- Zhenghao Liu, Xiaoyuan Yi, Maosong Sun, Liner Yang, and Tat-Seng Chua. 2021. Neural Quality Estimation with Multiple Hypotheses for Grammatical Error Correction. *arXiv preprint*. ArXiv:2105.04443 [cs].
- Sandeep Mathias and Pushpak Bhattacharyya. 2018. ASAP++: Enriching the ASAP Automated Essay Grading Dataset with Essay Attribute Scores. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Atsushi Mizumoto and Masaki Eguchi. 2023. Exploring the potential of using an AI language model for automated essay scoring. *Research Methods in Applied Linguistics*, 2(2):100050.
- Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.
- Christopher Ormerod, Amy Burkhardt, Mackenzie Young, and Sue Lottridge. 2023. Argumentation Element Annotation Modeling using XLNet. *arXiv preprint*. ArXiv:2311.06239 [cs].
- Christopher Michael Ormerod and Alexander Kwako. 2024. Automated Text Scoring in the Age of Generative AI for the GPU-poor. *arXiv preprint*. ArXiv:2407.01873.

- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer](#). Technical Report arXiv:1910.10683, arXiv. ArXiv:1910.10683 [cs, stat] type: article.
- Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. 2023. [CoEditIT: Text Editing by Task-Specific Instruction Tuning](#). *arXiv preprint*. ArXiv:2305.09857 [cs].
- Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. [A Simple Recipe for Multilingual Grammatical Error Correction](#). *arXiv preprint*. Number: arXiv:2106.03830 arXiv:2106.03830 [cs].
- Mark D. Shermis and Ben Hamner. 2013. [Contrasting State-of-the-Art Automated Scoring of Essays](#). pages 335–368. Publisher: Routledge Handbooks Online.
- Christian Stab and Iryna Gurevych. 2014. [Annotating Argument Components and Relations in Persuasive Essays](#). In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1501–1510, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.
- Christian Stab and Iryna Gurevych. 2017. [Parsing Argumentation Structures in Persuasive Essays](#). *Computational Linguistics*, 43(3):619–659.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. [Llama 2: Open Foundation and Fine-Tuned Chat Models](#). *arXiv preprint*. ArXiv:2307.09288 [cs].
- Masaki Uto and Yuto Uchida. 2020. [Automated Short-Answer Grading Using Deep Neural Networks and Item Response Theory](#). *AIED*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is All you Need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Changrong Xiao, Wenxing Ma, Sean Xin Xu, Kunpeng Zhang, Yufang Wang, and Qi Fu. 2024. [From Automation to Augmentation: Large Language Models Elevating Essay Scoring Landscape](#). *arXiv preprint*. ArXiv:2401.06431 [cs] version: 1.
- Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. [Parameter-Efficient Fine-Tuning Methods for Pretrained Language Models: A Critical Review and Assessment](#). *arXiv preprint*. ArXiv:2312.12148 [cs].
- Ruosong Yang, Jiannong Cao, Zhiyuan Wen, Youzheng Wu, and Xiaodong He. 2020. [Enhancing Automated Essay Scoring Performance via Fine-tuning Pre-trained Language Models with Combination of Regression and Ranking](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1560–1569, Online. Association for Computational Linguistics.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. [XLNet: Generalized Autoregressive Pretraining for Language Understanding](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. [A New Dataset and Method for Automatically Grading ESOL Texts](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

10 Appendix

Component ID	Component Class.	Stance	Error Detect (FCE)	Error Correct (JFLEG)
User: Task 1: Component Identification. List argumentative clauses without justification. {essay} Assistant: 1. {clause} 2. {clause} 3. {clause} . .	User: Read essay and label argumentative components: major claim / claim / premise using rubric. Essay: {essay}/ Rubric: {rubric} Component: {component} Assistant: {clause_type}	User: Given essay: {essay} Statement: {arg1} Claim: {arg2} Classify as: supports / attacks Assistant: The statement {arg1} relation the claim {arg2}	System: Spellchecker User: List: {arg1} Return only misspelled words as bullets: • incorrect word • incorrect word • ... Assistant: • {mistake 1} • {mistake 2} • ...	System: Spellchecker User: Sentence: {arg1} Return corrected sentence only (no explanation). Assistant: Corrected sentence: {sentence}

Table 6: All prompts used across tasks: component identification, classification, stance detection, error detection (FCE), and error correction (JFLEG).