

Argument Mining in Education: Exploring the Potential of Open-source Small LLMs for Argument Classification and Assessment

Lucile Favero^{1*}, Juan Antonio Pérez-Ortiz², Tanja Käser³ Nuria Oliver¹

¹ELLIS Alicante, Spain

²Universitat d'Alacant, Spain

³École Polytechnique Fédérale de Lausanne, EPFL, Switzerland

lucile@ellisalicante.org, japerez@dlsi.ua.es, tanja.kaeser@epfl.ch, nuria@ellisalicante.org

Abstract

Argument mining algorithms analyze the argumentative structure of essays, making them a valuable tool for enhancing education by providing targeted feedback about the students' argumentation skills. While current methods often use Encoder or Encoder-Decoder deep learning architectures, Decoder-only models remain largely unexplored, offering a promising research direction. In this paper, we propose leveraging open-source, small Large Language Models (LLMs) –such as Llama 3.1 8B– for argument mining through few-shot prompting and fine-tuning, to classify argument types and assess their quality in student essays. Their small size and open-source nature ensure greater accessibility, privacy, and computational efficiency, enabling schools and educators to adopt and deploy them locally. We empirically evaluate the proposed method using the “Feedback Prize – Predicting Effective Arguments” dataset, which contains essays from students in grades 6-12. We demonstrate that fine-tuned small LLMs outperform baseline methods in determining the argument types while achieving comparable performance to the baselines in assessing quality. This work illustrates the educational potential of small LLMs to deliver real-time, personalized feedback, fostering independent learning, and improved writing skills in students while maintaining low computational demands and prioritizing privacy.

1 Introduction

Writing essays can be challenging for students, as constructing a well-structured essay requires not only quality argument components but also cohesive connections between them (Scardamalia and Bereiter 1987). However, many students struggle to meet these requirements, often due to a lack of clear guidance on effective argumentation. High teacher-student ratios exacerbate this issue, limiting the individualized support that teachers can provide¹. To address this challenge, automatic argument mining has emerged as a promising solution, aiming to analyze the argumentative structure of essays and deliver targeted feedback to help students strengthen their arguments and overall essay structure (Lawrence and Reed 2020; Cabrio and Villata 2018).

*Corresponding Author.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

¹<https://www.unesco.org/en/articles/global-report-teachers-what-you-need-know>

Effective argument mining entails several subtasks, such as segmenting the essay into distinct argument components, classifying the type of these argument components, assessing their quality and establishing relationships between them (Gessler et al. 2021). These tasks can be performed sequentially or in parallel. State-of-the-art approaches in argument mining rely on Encoder or Encoder-Decoder deep neural network-based architectures (Arora, Rana, and Singh 2023). However, to date, Decoder-only models remain underexplored for certain subtasks, including argument type classification, suggesting a promising area for future exploration (Wachsmuth et al. 2024).

While high-performing argument mining methods have potential, they are often difficult to access and scale, especially in educational settings (Kashefi, Chan, and Somasundaran 2023). Automated Essay Scoring (AES) systems offer an alternative solution for grading essays, focusing on overall evaluation rather than on assessing individual arguments (Gao et al. 2024). However, little research has examined the integration of argument mining with argument quality assessment. Bridging this gap could give students more detailed feedback, helping them critically analyze their arguments and improve their writing skills (Ding, Bexte, and Horbach 2023). Our work aims to fill this gap.

We propose leveraging open-source, small Large Language Models (LLMs) –such as Llama 3.1, 8B parameters (Dubey et al. 2024)– to perform argument mining tasks, namely classifying the argument types and assessing their effectiveness in student essays. By means of few-shot prompting and fine-tuning, these models can execute these tasks locally on a standard laptop, ensuring privacy and maintaining computational efficiency. In experimental evaluations on the “Feedback Prize – Predicting Effective Arguments” dataset (Crossley et al. 2022, 2023), which includes essays of students from 6th to 12th grade, the proposed approach outperforms baseline methods in determining the argument types while achieving comparable performance to the baselines in assessing essay quality. Fig 1 provides a visual representation of our method's pipeline, outlining each stage of the process.

This paper is organized as follows: Section 2 reviews the most relevant literature, providing the background and context for our research. In Section 3, we describe our two experimental setups for conducting argument mining tasks:

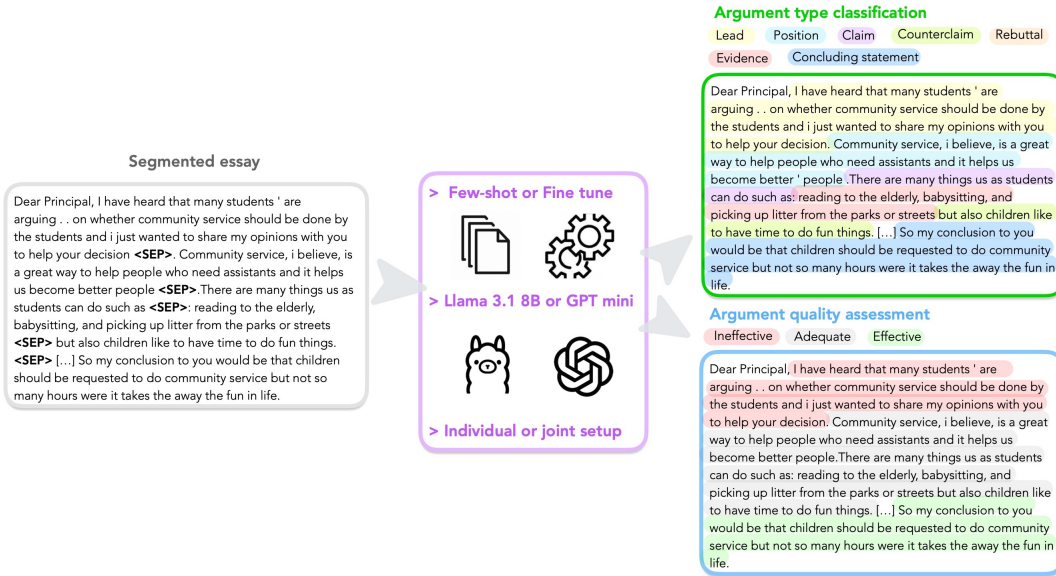


Figure 1: Overview of the proposed framework. Given a segmented essay as input, the goal is to classify the argument types and evaluate the argument quality by means of small LLMs (Llama 3.1 8B and GPT-4o mini) and using two approaches: few-shot prompting or fine-tuning. The tasks are carried out either individually or jointly.

(1) few-shot prompt tuning and (2) fine-tuning of an open-source pre-trained LLM. Section 4 presents and analyses our experimental results, including comparisons with baselines. Finally, Section 5 provides a discussion of the findings, followed by a conclusion in Section 6, highlighting key takeaways and suggesting directions for future research. Our code is available at https://anonymous.4open.science/t/AAAI_25_AI4ED-ED21.

2 Related work

2.1 Argument mining

Argument mining is a complex field that aims to identify, classify, and analyze argumentative structures within text (Lawrence and Reed 2020), drawing inspiration from frameworks, such as Toulmin’s model of argumentation (Toulmin 2003). Argument mining involves numerous sub-tasks (Arora, Rana, and Singh 2023), including argument detection, classification, assessment, and relation prediction, making end-to-end solutions particularly challenging (Cabrio and Villata 2018). Thus and despite its importance, few research works address the entire pipeline in argument mining due to its complexity and the diversity in required methodologies (Cao 2023; Bao et al. 2022; Morio et al. 2022).

State-of-the-art methods in argument mining typically rely on deep neural networks (Arora, Rana, and Singh 2023). Recently, advancements in Large Language Models (LLMs) have pushed the field forward. For instance, T5 has been applied effectively to argument mining tasks (Kawarada et al. 2024), while models like Longformer (Ding, Bexte, and Horbach 2023) and BERT-based approaches (Kashefi,

Chan, and Somasundaran 2023) have demonstrated competitive performance across various subtasks. More complex systems provide end-to-end solutions by combining models like BART with prompting and graph-based approaches (Sun et al. 2024b), or by leveraging graph prefix tuning to enhance discourse-level understanding (Sun et al. 2024a). Recently, (Gorur, Rago, and Toni 2024) demonstrates that prompt-tuned, open-source models like Llama-2 and Mistral can outperform state-of-the-art RoBERTa-based baselines in identifying agreement and disagreement relations among arguments. However, to the best of our knowledge, no research has explored to date the use of open-source, small LLMs for the combined tasks of argument classification and quality assessment. In this paper, we aim to fill this gap.

2.2 Educational multi-task argument mining

Educational multi-task argument mining focuses on extracting, classifying, and evaluating arguments in student essays—a challenging problem due to the noisy, resource-constrained nature of student writing (Kashefi, Chan, and Somasundaran 2023). Beyond the tasks of argument segmentation and classification, assessing the quality of arguments is essential for evaluating their persuasiveness and coherence (Wachsmuth et al. 2024). Existing approaches, such as Longformer-based classification methods (Ding, Bexte, and Horbach 2022) and graph-based frameworks (Marro, Cabrio, and Villata 2022), have made contributions to this area.

Providing meaningful feedback from such analyses is particularly impactful in educational contexts. Actionable feedback enables students and educators to identify strengths and areas for improvement, with standardized scoring systems

serving as valuable tools to guide learning and enhance outcomes (Cabrio and Villata 2018). Moreover, incorporating discourse-level features has been shown to improve performance by offering deeper insights into argument structures (Deshpande et al. 2023).

In this paper, we make several contributions to this domain. First, we address the gap in leveraging small, open-source LLMs for argument mining, combining argument type classification and quality assessment. Second, we propose a computationally efficient and privacy-preserving approach, enabling local analysis on standard devices through fine-tuning and few-shot prompting of the LLMs. Finally, by evaluating our approach on a benchmark dataset of student essays, we demonstrate its ability to deliver actionable feedback on a local computer, fostering improved writing skills for students grades 6-12 while preserving privacy. Our method advances argument mining in resource-constrained educational settings and highlights the transformative potential of LLMs in personalized education.

3 Method

3.1 Tasks: Argument classification and assessment

Given the segmentation of a human-written essay into argument components or *segments*, the tasks to achieve by our method consist of (1) classifying the type of each segment according to seven different classes, namely *Lead*, *Position*, *Claim*, *Counterclaim*, *Rebuttal*, *Evidence*, *Concluding Statement* and (2) assessing the quality of each segment according to the following scores: *Ineffective*, *Adequate*, and *Effective*. These classification and scoring tasks can be executed independently or within a joint framework. In the remainder of this paper, we will refer to these configurations as the *individual setup* and the *joint setup*, respectively.

Our aim is to explore the potential of open-source, small LLMs –namely Llama 3.1 8B²– to perform these tasks by means of few-shot prompting and fine-tuning.

3.2 Few-shot prompting

The first technique to perform the desired tasks entails prompt-tuning the LLM to perform argument type classification and quality assessment. In addition to Llama 3.1 8B (Dubey et al. 2024), we also run this process with GPT-4o mini (Achiam et al. 2023) to compare the performance of an open-source model with a closed-source one.

The prompt is built as follows:

1. **Few-shot examples** are given in the format depicted by the examples included in Table 11 and Table 12 (in the supplementary material) depending on the task. The label of the type of argument or its quality is added between each argument segment of an essay. The example essays are extracted from the training set split used in (Ding, Bexte, and Horbach 2023) and described in Section 3.3.

²Section 7.6 in the supplementary material includes the experimental results in argument classification with three additional open-source small LLMs using three-shot prompting, namely Llama 3.2 2B, Qwen 2.5 7B and Gemma 2 9B.

2. **The already segmented essay** is provided following the format exemplified in Table 10, in the supplementary material: a separator *SEP* is added between each segment.
3. **The query** specifies the role of the LLM and the overall instructions for the task. The detailed formulations for the argument type classification and quality assessment tasks are provided as per Table 1.
4. **The output requirement** as follows: “For the *argument component*, identify its [type] or [quality]. Provide your output as a JSON object with the key: [TYPE] and/or [QUALITY].”
5. **The specific segment** of the essay that the LLM is asked to analyze.

Table 1: Queries used in both LLM few-shot prompting (Section 3.2) and fine-tuning inference (Section 3.3) for the argument type classification and quality assessment tasks.

Type	You are a very strict AI evaluator specializing in detecting the type of argument components in essays. The argument types are as follows: <ul style="list-style-type: none"> - <i>Lead</i>: An introduction that begins with a statistic, quotation, description, or other device to grab the reader’s attention and point toward the thesis. - <i>Position</i>: An opinion or conclusion on the main question. - <i>Claim</i>: A statement that supports the position. - <i>Counterclaim</i>: A statement that opposes another claim or provides an opposing reason to the position. - <i>Rebuttal</i>: A statement that refutes a counterclaim. - <i>Evidence</i>: Ideas or examples that support claims, counterclaims, or rebuttals. - <i>Concluding Statement</i>: A statement that restates the claims and summarizes the argument.
Quality	You are a very strict AI evaluator specializing in detecting the quality of argument components in essays. Each component should be rated as <i>Ineffective</i> , <i>Adequate</i> , or <i>Effective</i> , with <i>Effective</i> being the highest quality.

Examples of the expected output format are presented in Table 14 in the supplementary material.

Note that asking the LLM to classify the entire text at once and output a list of predicted labels can lead to mismatches, where labels may not correspond correctly to their intended segments. To simplify this task and make it manageable for smaller LLMs, we ask the LLM to classify one segment per prompt. However, the full essay is still provided as context, given that the classification and quality of each segment depend on the essay’s overall argumentation strategy.

Also note that the specific structure, content and phrasing of each prompt have been meticulously designed after an intensive effort of prompt engineering. Any changes to the sequence or omission of parts significantly affect both the accuracy and the format of the output.

An alternative approach to prompt-tuning the LLM would involve asking the LLM to replace each separator with its corresponding label (type, quality, or both), as demonstrated in the fine-tuning process described in the next section. With this method, the output should exactly replicate the original essay, with classified labels inserted at the appropriate

points between segments. However, this approach did not work well for smaller LLMs as these models struggle to reproduce the input text word-for-word and often introduce errors such as adding extra words, omitting parts of the text, or inserting additional separators, which leads to poor performance and unreliable classification outcomes. For output examples, see Section 7.1 in the supplementary material.

3.3 Fine-tuning

The second approach involves fine-tuning an LLM for argument type classification and quality assessment, both as an individual and as a joint task. We fine-tuned the Llama 3.1 8B parameter model (Dubey et al. 2024) on Google Colab using an A100 GPU, and performed three separate runs of each task: (1) argument type classification, (2) argument quality assessment, and (3) both argument type classification and quality assessment together.

Data Fine-tuning requires the use of a dataset. In our study, we used a collection of argumentative essays written by U.S. students in grades 6-12, annotated by expert raters. The dataset is derived from the Kaggle competition “Feedback Prize - Predicting Effective Arguments,”³ which constitutes a subset of the PERSUADE 2.0 Corpus. This subset includes approximately 6,900 essays from a total of 26,000 argument components, representing just over a quarter of the corpus. The essays were selected to achieve a balanced distribution of discourse elements across varying levels of effectiveness (Crossley et al. 2022, 2023). The dataset encompasses the seven argument component types derived from Toulmin’s argumentation model (Toulmin 2003): *Lead*, *Position*, *Claim*, *Counterclaim*, *Rebuttal*, *Evidence*, and *Concluding Statement*; along with a quality assessment for each argument component: *Ineffective*, *Adequate*, *Effective*.

For fine-tuning, we used the split provided in (Ding, Bexte, and Horbach 2023) consisting of 3,353 essays (*i.e.*, 29,440 argument segments) for the training set and 419 essays (*i.e.*, 3,614 argument segments) for the validation set. Detailed statistics of this dataset split can be found in (Ding, Bexte, and Horbach 2022) and (Ding, Bexte, and Horbach 2023).

Baseline To establish a comparative upper bound, we also fine-tuned GPT-4o mini. Further details on the fine-tuning of this model can be found in Section 7.3 of the supplementary material.

Fine-tune setting We performed the fine-tuning using the *SFTTrainer* module from the *TRL* library⁴. We employed *Unsloth*⁵ to optimize performance and reduce memory usage. It is an open-source Python framework designed to enhance the efficiency of fine-tuning LLMs by implementing custom Triton kernels and a manual backpropagation engine. Additionally, we incorporated Low-Rank Adaptation

(LoRA) (Hu et al. 2021) and Quantized Low-Rank Adaptation (QLoRA) (Dettmers et al. 2024) to further reduce memory demands and improve the fine-tuning speed. An early stopping criterion was applied to optimize performance and reduce memory usage. The specific hyperparameters used in the fine-tuning can be found in Section 7.3 of the supplementary material.

Input sequence and target The input sequence was composed of the following elements:

1. **The query**, was the same query as the one used for few-shot prompting (see Section 3.2 and Table 1).
2. **The output requirement** consisting of the following text: “Given the following essay, replace each instance of SEP with the argument type [the quality] of the segment immediately preceding it.”
3. **The already segmented essay**, following the same format as the one used for few-shot prompting and exemplified in Table 10.

The target of the fine-tuning was in the same format as the few-shot examples. Therefore, it was composed of an essay segmented into arguments, with the label of the type of argument and/or its quality added between each of the segments. Examples of this format can be found in Table 11, Table 12, and Table 13, in the supplementary material, depending on the task.

Inference The fine-tuned Llama 3.1 8B was run on an Apple M1 Pro laptop with 32 GB RAM using Ollama⁶, an open-source framework that enables users to run, create, and share LLMs locally on their machines. Thanks to the early stopping method, the fine-tuning stopped after 890 steps in the argument type classification task, 840 steps in the argument quality assessment task, and 540 steps in the joint setup. We did not provide any examples demonstrating how to perform the task or how to specify the output format. To evaluate performance, we used the same test set as in (Ding, Bexte, and Horbach 2023). Fine-tuning is expected to enhance the model’s ability to generate outputs that closely mirror the input essay and conform better to the specified output format than the no fine-tuned models.

4 Evaluation

4.1 Dataset

We performed all our evaluations on the test set of the “Feedback Prize - Predicting Effective Arguments,”⁷ datasets. We use the same test set employed in (Ding, Bexte, and Horbach 2023), which is composed of 419 essays, containing a total of 3,711 argument segments. Detailed statistics of this dataset split can be found in (Ding, Bexte, and Horbach 2022) and (Ding, Bexte, and Horbach 2023).

For the argument type classification task, the arguments had to be classified into one of seven types: *Lead*, *Position*, *Claim*, *Counterclaim*, *Rebuttal*, *Evidence*, and *Concluding*

³<https://www.kaggle.com/competitions/feedback-prize-effectiveness/data>

⁴https://huggingface.co/docs/trl/sft_trainer

⁵<https://github.com/unslothai/unsloth>

⁶<https://github.com/ollama/ollama>, <https://ollama.com>

⁷<https://www.kaggle.com/competitions/feedback-prize-effectiveness/data>

Statement. In the case of the quality assessment task, the possible values are: *Ineffective, Adequate, Effective.*

4.2 Performance metrics

To assess the efficiency of the proposed methods, we report the following metrics due to the imbalance in label distribution.

Metrics per label, namely precision, recall, and F1 score for each individual label to have an in-depth look at the classifier’s performance on a label basis.

Multi-label confusion matrix to provide a detailed breakdown of model performance across argument categories, highlighting both accurately predicted cases (along the diagonal) and common misclassifications. The values in the matrix give insights into which argument types or quality assessment labels the proposed method distinguishes effectively and where it struggles.

The macro-averaged F1 score, which is the mean of the F1 scores for each label, treating all labels equally, thereby providing a measure of overall performance across all labels without considering label imbalance. This metric is a standard in the argument mining community. For n classes, the macro-averaged F1 score is $F_1 = \frac{1}{n} \sum_{i=1}^n F_{1,i}$, where the F1 score of class i is given by:

$$F_{1,i} = \frac{2 \cdot \text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i},$$

with Precision_i and Recall_i representing the precision and recall scores of class i .

4.3 Baselines

We compare our results with the Longformer-based approach proposed in (Ding, Bexte, and Horbach 2023), as it is the only method identified from the literature that utilizes this dataset for the same tasks and takes as input the segmented essays. Their technique employs the Longformer model, which is specifically designed for long-text classification, and it is trained to perform both tasks simultaneously. Additionally, as mentioned earlier, we compare our results with GPT-4o mini (Achiam et al. 2023), both vanilla and fine-tuned versions, to have a sense of the upper performance bound achieved by a commercial model.

4.4 Results

4.4.1. Few-shot prompting Table 2 and Table 3 present the results of the comparative analysis of the performance of Llama 3.1 8B and GPT-4o mini on the argument type classification task and quality assessment task, respectively. The tables detail the accuracy and macro-averaged F1 as the number of few-shot examples is increased, with values: 0, 3, 6, 12, and 20 examples. Results are averaged over two runs. As reflected in Table 15 and Table 16 in the supplementary material, the standard deviations of the performance across runs are negligible.

As seen in the Tables, GPT-4o mini consistently outperforms Llama 3.1 8B. Specifically, GPT-4o mini achieves the highest performance in the three-shot setting for the argument type classification task and the twelve-shot setting for

Table 2: Performance metrics (accuracy (ACC) and macro-averaged F1 (Macro)) for the argument type classification task when increasing the number of few-shot examples, comparing the models Llama 3.1 8B and GPT-4o mini. Bolded values indicate the best performance for each model based on the given metric. Results are averaged over two runs with negligible standard deviations reported in the supplementary material.

Type	Model	Shots				
		0	3	6	12	20
ACC	Llama 3.1	0.42	0.43			
	GPT-4o mini	0.62	0.63	0.62	0.63	0.63
Macro	Llama 3.1	0.38	0.40			
	GPT-4o mini	0.60	0.60	0.58	0.59	0.60

Table 3: Performance metrics (accuracy (ACC) and macro-averaged F1 (Macro)) for the quality assessment task when increasing the number of few-shot examples, comparing the models Llama 3.1 8B and GPT-4o mini. Bolded values indicate the best performance for each model based on the given metric. Results are averaged over two runs with negligible standard deviations reported in the supplementary material.

Type	Model	Shots				
		0	3	6	12	20
ACC	Llama 3.1	0.54	0.48			
	GPT-4o mini	0.57	0.58	0.59	0.59	0.58
Macro	Llama 3.1	0.44	0.40			
	GPT-4o mini	0.52	0.53	0.53	0.54	0.53

the quality assessment task. Llama 3.1 8B performs best in the three-shot setting for the argument type classification and in the zero-shot setting or the quality assessment task.

The confusion matrices and the performance evaluation metrics for Llama 3.1 8B with best shot prompting are included in the supplementary material both for the argument type classification (see Table 17 and Table 18) and for the quality assessment (see Table 19 and Table 20) tasks. Furthermore, Section 7.6 in the supplementary material includes the experimental results in argument classification with three additional open-source small LLMs using three-shot learning, namely Llama 3.2 2B, Qwen 2.5 7B and Gemma 2 9B.

4.4.2. Fine-tuned models Table 4 and Table 5 provide a comparison of the performance (accuracy and F1 macro) of the following models: the Longformer (Ding, Bexte, and Horbach 2023); the best performing models using few-shot prompting, namely Llama 3.1 8B with zero and three-shot, GPT-4o mini with three and twelve-shot and GPT-4o with zero-shot; and the fine-tuned models with 3,711 argument segments, namely the fine-tuned Llama 3.1 8B both individually and jointly, and the fine-tuned GPT-4o jointly.

The aim of this experiment was to evaluate the performance of a fine-tuned, small, open-source LLM –Llama 3.1 8B– when compared to a larger, commercial LLM –GPT-4o– and a specialized models –the Longformer, which has been designed to perform argument mining tasks. We also wanted to shed light on the effectiveness of few-shot prompt-

Table 4: Performance metrics (accuracy (ACC) and the macro-average F1 score (F1 macro) in the argument type classification task of a variety of models. The best results are highlighted in bold and the second best results are underlined.

	Model	ACC	F1 macro
	Longformer	<u>0.82</u>	0.71
few-shot	Llama 3.1, 3-shot	0.43	0.40
	GPT-4o mini, 3-shot	0.63	0.60
	GPT-4o, 0-shot	0.67	0.68
fine-tuned	Llama FT, indiv.	0.76	0.74
	Llama FT, joint	0.81	<u>0.78</u>
	GPT-4o mini FT, joint	0.86	0.84

Table 5: Performance metrics (accuracy (ACC) and the macro-average F1 score (F1 macro) in the quality assessment task of a variety of models. The best results are highlighted in bold and the second best results are underlined.

	Model	ACC	F1 macro
	Longformer	0.69	0.51
few-shot	Llama 3.1, 0-shot	0.54	0.44
	GPT-4o mini, 12-shot	0.59	<u>0.54</u>
	GPT-4o, 0-shot	0.47	0.41
fine-tuned	Llama FT, indiv.	0.54	0.51
	Llama FT, joint	0.56	0.51
	GPT-4o mini FT, joint	<u>0.61</u>	0.55

ing vs fine-tuning small LLMs. Hence, we also include in the tables the performance of the best performing models using few-shot prompting.

In the case of the argument type classification task, the performance of the fine-tuned models is superior to the performance of the rest of LLM models across all metrics. Interestingly, the fine-tuned Llama 3.1 8B in the joint setup achieved very competitive performance which is higher than in the individual setup, outperforming the Longformer in terms of F1 macro and yielding comparable results in accuracy. This suggests that joint fine-tuning enhances the model’s ability to accurately classify argument types. The fine-tuned GPT-4o mini is the best performing model overall, with significant performance gains over the Longformer baseline.

Regarding the quality assessment task, GPT-4o mini yields the best F1 macro, while the fine-tuned Llama 3.1 8B models achieve the same F1 macro performance as the Longformer baseline. However, in terms of accuracy, neither the best-performing GPT-4o model nor the fine-tuned Llama 3.1 8B models are able to achieve the Longformer’s results.

Label-level performance analysis To better understand the previously reported performance figures, we provide the results at the label level. Table 6 and Table 8 depict the precision, recall and F1 score of the best performing Llama 3.1 8B model (*i.e.*, fine-tuned on the joint setup). Table 7 and Table 9 contain the corresponding confusion matrices. All results are averaged over four runs, with negligible standard

deviations which are included in the supplementary material.

With respect to the argument type classification task, the worst performance in terms of precision and F1-score corresponds to *Evidence* and *Lead* whereas the best performance can be found when classifying *Counterclaim* and *Position* for precision and *Claim* and *Rebuttal* for F1-score. Regarding recall, the LLM excels in the case of *Claim* and *Lead* and has difficulties with *Counterclaim* and *Concluding statement*.

Table 7 contains the confusion matrix which illustrates the model’s tendency to misclassify certain argument types. Misclassifications of *Evidence* as *Claim* and the other way round align with *Evidence*’s and *Claim* relatively lower precision. Similarly, *Lead* is often misclassified as *Position*, which may explain the low precision in classifying *Lead*. The high precision but low recall for *Counterclaim* is due to frequent misclassifications of counterclaims with other argument types, and particularly with *Claim*, which makes intuitive sense as the language of both claims and counterclaims can be similar, often structured as declarative sentences without clear markers of opposition. Implicit counterclaims, indirect challenges, or arguments that share underlying assumptions are also hard to correctly detect. Furthermore, LLMs focus more on surface-level coherence rather than deep logical analysis, making it difficult to distinguish between opposing claims when both involve similar terminology or evidence. Lastly, without clear context about who holds which position, the LLM may misinterpret the stance of the arguments, leading to confusion between the two.

Type	Precision	Recall	F1-score
Claim	0.78	0.93	0.84
Conclud.	0.80	0.81	0.80
C claim	0.88	0.75	0.81
Evidence	0.53	0.83	0.65
Lead	0.58	0.91	0.71
Position	0.83	0.84	0.83
Rebuttal	0.79	0.90	0.84

Table 6: Performance evaluation (precision, recall and F1-score per label) in the argument type classification task using the fine-tuned Llama 3.1 8B model on the joint setup with the test set of 3,711 argument segments. Results averaged over four runs.

Regarding the argument quality assessment task, in Table 8 we observe that the best performance corresponds to the *Adequate* label with the highest precision (0.62) and recall (0.66), indicating the model can reliably identify adequate segments without many false positives. Conversely, the most difficult label for the LLM is *Ineffective* with low precision (0.37) and recall (0.38), showing that the model struggles both in finding true positives and avoiding false positives for effective segments. The confusion matrix depicted in Table 9 provides additional detail about the misclassification of the different quality assessment labels. Note that *Adequate* is the most common label in the dataset and *Ineffective* is the least common with more than three times fewer occurrences than *Adequate*, which partially explains

Type	Lead	Pos	Clai	C Clai	Reb	Evid	Ccl
Lead	158.00	36.00	4.00	0.00	5.75	0.00	0.00
Pos	8.00	308.50	55.75	0.25	0.00	5.25	10.25
Clai	1.00	19.50	986.00	8.75	1.50	102.25	5.50
C Clai	0.00	0.50	60.75	86.00	0.25	13.50	0.00
Reb	0.00	2.50	9.25	15.00	61.25	27.75	4.00
Evid	3.00	4.50	168.00	6.25	3.50	934.50	7.25
Ccl	0.50	8.25	34.50	0.25	7.75	25.50	255.75

Table 7: Confusion matrix for the classification of the type of argument using the fine-tuned Llama 3.1 8B model on the joint setup with the test set of 3,711 argument segments. Results averaged over four runs. The argument types are: *Lead*, *Position*, *Claim*, *Counterclaim*, *Rebuttal*, *Evidence*, and *Concluding Statement*.

the results.

Quality	Precision	Recall	F1-score
Ineffective	0.37	0.38	0.37
Adequate	0.62	0.66	0.64
Effective	0.55	0.47	0.50

Table 8: Performance evaluation (precision, recall and F1-score per label) in the argument quality assessment task using the fine-tuned Llama 3.1 8B model on the joint setup with the test set of 3,711 argument segments. Results averaged over four runs.

Quality	Ineffective	Adequate	Effective
Ineffective	223.75	320.00	64.00
Adequate	348.75	1269.75	427.75
Effective	19.75	333.75	428.50

Table 9: Confusion matrix for the quality assessment task using the fine-tuned Llama 3.1 model on the joint setup with the test set of 3,711 argument segments. Results averaged over four runs. The quality scores, sorted in increasing order, are: *Ineffective*, *Adequate*, and *Effective*.

5 Discussion

In this paper, we have explored the potential of small, open-source LLMs –namely Llama 3.1 8B– to perform two argument mining tasks in an educational setting: argument type classification and argument quality assessment. We have experimented with both few-shot prompting and fine-tuning, and we have compared the performance of Llama 3.1 8B with that of commercial LLMs (GPT-4o mini) and state-of-the-art methods (Longformer).

From extensive experiments with over 3,700 argument segments corresponding to essays written by students in grades 6-12, we draw several findings.

First, fine-tuned models significantly overperform few-shot prompting both in the case of open-source and commercial models.

Second, model performance in the tasks of interest does not necessarily increase with the number of shots when performing few-shot prompting, which is consistent with what

has been reported in the literature (Liu et al. 2024). This finding is particularly evident in the case of Llama 3.1 8B, which exhibits the best performance in tasks with zero and three-shot prompting. The deterioration in performance with the number of shots is probably due to the complexity of longer prompts which seemed too hard for the model to make sense of.

Third, small LLMs had trouble analyzing entire essays, often being unable to perform the tasks and/or making numerous mistakes⁸. To address this limitation, the essays were segmented in arguments.

Fourth, fine-tuning small LLMs (both Llama 3.1 8B and GPT-4o mini) allowed for essay-level assessments which surpassed or reached benchmark-level performance, overcoming the limits of few-shot prompting.

Fifth, joint fine-tuning setups where both tasks were carried out at the same time yielded better results than individual setups where the tasks were performed independently, showing a strong link between classifying argument types and assessing their quality (Crossley et al. 2023).

Sixth, the automatic classification of the type of argument seems to be an easier task than the assessment of the quality of the arguments, likely due to difficulties in creating high-quality and consistent ground truth quality assessments across essays (Wachsmuth et al. 2024)⁹.

Finally and most importantly, this study highlights the potential of open-source, small LLMs, running locally on personal computers, to support students in the development of their essay writing skills. Our framework prioritizes privacy and accessibility, addressing the challenge of developing efficient models for local use without consuming too many resources (Kashefi, Chan, and Somasundaran 2023).

6 Conclusion and future work

In this paper, we have presented a study of the potential of small, open-source LLMs for argument mining, investigating their effectiveness in both few-shot prompting and fine-tuning setups. Fine-tuning proved especially valuable, achieving competitive performance in essay-level assessments. Our findings underscore the importance of joint setups for improved argument type classification. By focusing on models running locally on a personal computer, our research promotes privacy, accessibility, and resource efficiency, illustrating the potential of open-source, small LLMs as a promising tool for educational applications.

However, our work is not exempt from limitations that we plan to address in future work. First, we have carried out all experiments with pre-segmented essays into arguments. For broader, real-world applicability, we plan to develop a system capable of performing segmentation locally and efficiently before carrying out the argument type classification and quality assessment. Second, we have identified poor quality in the annotated data used to fine-tune and evaluate the models. Hence, we plan to improve the annotation quality and create a more reliable dataset to be shared with the research community.

⁸See 7.1 in the supplementary material for examples.

⁹See 7.4 in the supplementary material for a discussion.

Acknowledgements

L.F. and N.O. have been partially supported by a nominal grant received at the ELLIS Unit Alicante Foundation from the Regional Government of Valencia in Spain (Resolución de la Conselleria de Innovación, Industria, Comercio y Turismo, Dirección General de Innovación). L.F. has also been partially funded by a grant from the Banc Sabadell Foundation.

References

- Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Arora, S.; Rana, A.; and Singh, A. 2023. Argument Mining: A Categorical Review. In *Modern Electronics Devices and Communication Systems: Select Proceedings of MEDCOM 2021*, 353–367. Springer.
- Bao, J.; He, Y.; Sun, Y.; Liang, B.; Du, J.; Qin, B.; Yang, M.; and Xu, R. 2022. A generative model for end-to-end argument mining with reconstructed positional encoding and constrained pointer mechanism. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 10437–10449.
- Cabrio, E.; and Villata, S. 2018. Five years of argument mining: A data-driven analysis. In *IJCAI*, volume 18, 5427–5433.
- Cao, L. 2023. AutoAM: An End-To-End Neural Model for Automatic and Universal Argument Mining. In *International Conference on Advanced Data Mining and Applications*, 517–531. Springer.
- Crossley, S. A.; Baffour, P.; Tian, Y.; Franklin, A.; Benner, M.; and Boser, U. 2023. A large-scale corpus for assessing written argumentation: PERSUADE 2.0. Available at SSRN 4795747.
- Crossley, S. A.; Baffour, P.; Tian, Y.; Picou, A.; Benner, M.; and Boser, U. 2022. The persuasive essays for rating, selecting, and understanding argumentative and discourse elements (PERSUADE) corpus 1.0. *Assessing Writing*, 54: 100667.
- Deshpande, D.; Sourati, Z.; Ilievski, F.; and Morstatter, F. 2023. Contextualizing argument quality assessment with relevant knowledge. *arXiv preprint arXiv:2305.12280*.
- Dettmers, T.; Pagnoni, A.; Holtzman, A.; and Zettlemoyer, L. 2024. QLoRA: Efficient finetuning of quantized LLMs. *Advances in Neural Information Processing Systems*, 36.
- Ding, Y.; Bexte, M.; and Horbach, A. 2022. Don't drop the topic-the role of the prompt in argument identification in student writing. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, 124–133.
- Ding, Y.; Bexte, M.; and Horbach, A. 2023. Score it all together: A multi-task learning study on automatic scoring of argumentative essays. In *Findings of the Association for Computational Linguistics: ACL 2023*, 13052–13063.
- Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Gao, R.; Merzdorf, H. E.; Anwar, S.; Hipwell, M. C.; and Srinivasa, A. 2024. Automatic assessment of text-based responses in post-secondary education: A systematic review. *Computers and Education: Artificial Intelligence*, 100206.
- Gessler, L.; Behzad, S.; Liu, Y. J.; Peng, S.; Zhu, Y.; and Zeldes, A. 2021. DisCoDisCo at the DISRPT2021 shared task: A system for discourse segmentation, classification, and connective detection. *arXiv preprint arXiv:2109.09777*.
- Gorur, D.; Rago, A.; and Toni, F. 2024. Can Large Language Models perform Relation-based Argument Mining? *arXiv preprint arXiv:2402.11243*.
- 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. *arXiv preprint arXiv:2106.09685*.
- Kashefi, O.; Chan, S.; and Somasundaran, S. 2023. Argument detection in student essays under resource constraints. In *Proceedings of the 10th Workshop on Argument Mining*, 64–75.
- Kawarada, M.; Hirao, T.; Uchida, W.; and Nagata, M. 2024. Argument Mining as a Text-to-Text Generation Task. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2002–2014.
- Lawrence, J.; and Reed, C. 2020. Argument mining: A survey. *Computational Linguistics*, 45(4): 765–818.
- Liu, N. F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; and Liang, P. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12: 157–173.
- Marro, S.; Cabrio, E.; and Villata, S. 2022. Graph embeddings for argumentation quality assessment. In *EMNLP 2022-Conference on Empirical Methods in Natural Language Processing*.
- Morio, G.; Ozaki, H.; Morishita, T.; and Yanai, K. 2022. End-to-end argument mining with cross-corpora multi-task learning. *Transactions of the Association for Computational Linguistics*, 10: 639–658.
- Scardamalia, M.; and Bereiter, C. 1987. Knowledge telling and knowledge transforming in written composition. *Advances in applied psycholinguistics*, 2: 142–175.
- Sun, Y.; Chen, G.; Yang, C.; Bao, J.; Liang, B.; Zeng, X.; Yang, M.; and Xu, R. 2024a. Discourse Structure-Aware Prefix for Generation-Based End-to-End Argumentation Mining. In *Findings of the Association for Computational Linguistics ACL 2024*, 11597–11613.
- Sun, Y.; Wang, M.; Bao, J.; Liang, B.; Zhao, X.; Yang, C.; Yang, M.; and Xu, R. 2024b. PITA: Prompting Task Interaction for Argumentation Mining. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5036–5049.

Toulmin, S. E. 2003. *The uses of argument*. Cambridge university press.

Wachsmuth, H.; Lapesa, G.; Cabrio, E.; Lauscher, A.; Park, J.; Vecchi, E. M.; Villata, S.; and Ziegenbein, T. 2024. Argument Quality Assessment in the Age of Instruction-Following Large Language Models. *arXiv preprint arXiv:2403.16084*.

7 Supplementary material

7.1 Input and output format

Table 10 presents an example of essay segmentation format. Table 11, Table 12 and Table 13 present examples of the

Table 10: An example of essay segmentation format.

Hi, i'm Isaac, i'm going to be writing about how this face on Mars is a natural landform or if there is life on Mars that made it. The story is about how NASA took a picture of Mars and a face was seen on the planet. NASA doesn't know if the landform was created by life on Mars, or if it is just a natural landform. <SEP>. On my perspective, I think that the face is a natural landform because I dont think that there is any life on Mars. In these next few paragraphs, I'll be talking about how I think that is is a natural landform <SEP>I think that the face is a natural landform because there is no life on Mars that we have discovered yet <SEP> [...] Though people were not satified about how the landform was a natural landform, in all, we new that aliens did not form the face. I would like to know how the landform was formed. we know now that life on Mars doesn't exist. <SEP>

few-shot and fine-tuned output format used for the argument type classification and quality assessment tasks in the individual and join setups, respectively. Note that the tables retain spelling errors present in the original essay, as it directly reflects the source text.

Table 14 shows examples of the expected output format of the LLM prompt tuning for the argument type and quality classification tasks.

7.2 Details of few-shot prompting

Standard deviation of the few-shot experiments Table 15 and Table 16 depict the standard deviation of two runs of the performance metrics (accuracy (ACC) and macro-averaged F1 (Macro)) for the argument type classification and quality assessment tasks, respectively across when increasing the number of few-shot examples. Comparison of Llama 3.1 8B and GPT-4o mini.

Incorrect outputs generated with Llama 3.1 8B We include below several examples of incorrect outputs generated with Llama 3.1 8B for the argument type task, along with their corresponding input. Note how in the first output, Llama 3.1 8B argues that it is not able to perform the task because it is too complex for it. In the second output, a part of

Table 11: An example of the few-shot and fine-tuned output format used for the argument type classification task.

Hi, i'm Isaac, i'm going to be writing about how this face on Mars is a natural landform or if there is life on Mars that made it. The story is about how NASA took a picture of Mars and a face was seen on the planet. NASA doesn't know if the landform was created by life on Mars, or if it is just a natural landform. <Lead>. On my perspective, I think that the face is a natural landform because I dont think that there is any life on Mars. In these next few paragraphs, I'll be talking about how I think that is is a natural landform <Position>I think that the face is a natural landform because there is no life on Mars that we have discovered yet <Claim> [...] Though people were not satified about how the landform was a natural landform, in all, we new that aliens did not form the face. I would like to know how the landform was formed. we know now that life on Mars doesn't exist. <Concluding Statement>

Table 12: An example of the few-shot and fine-tuned output format used for the quality assessment task.

Hi, i'm Isaac, i'm going to be writing about how this face on Mars is a natural landform or if there is life on Mars that made it. The story is about how NASA took a picture of Mars and a face was seen on the planet. NASA doesn't know if the landform was created by life on Mars, or if it is just a natural landform. <Adequate>. On my perspective, I think that the face is a natural landform because I dont think that there is any life on Mars. In these next few paragraphs, I'll be talking about how I think that is is a natural landform <Adequate>I think that the face is a natural landform because there is no life on Mars that we have discovered yet <Adequate> [...] Though people were not satified about how the landform was a natural landform, in all, we new that aliens did not form the face. I would like to know how the landform was formed. we know now that life on Mars doesn't exist. <Ineffective>

the essay is missing: "The Electoral College is also a modern sense", and Llama 3.1 8B has omitted the classifications of some arguments.

- **Input:** "#QUERY: You are a very strict AI evaluator specializing in detecting the type of argument in essays. The argument types are as follows: - <Lead >: An introduction that begins with a statistic, quotation, description, or other device to grab the reader's attention and point toward the thesis. - <Position >: An opinion or conclusion on the main question. - <Claim >: A statement that supports the position. - <Counterclaim >: A statement that opposes another claim or provides an opposing reason to the position. - <Rebuttal >: A statement that refutes

Table 13: An example of the few-shot and fine-tuned output format used for the argument type classification and quality assessment tasks.

Hi, i'm Isaac, i'm going to be writing about how this face on Mars is a natural landform or if there is life on Mars that made it. The story is about how NASA took a picture of Mars and a face was seen on the planet. NASA doesn't know if the landform was created by life on Mars, or if it is just a natural landform. <Lead, Adequate >. On my perspective, I think that the face is a natural landform because I dont think that there is any life on Mars. In these next few paragraphs, I'll be talking about how I think that is is a natural landform <Position, Adequate>I think that the face is a natural landform because there is no life on Mars that we have discovered yet <Claim, Adequate> [...] Though people were not satisfied about how the landform was a natural landform, in all, we new that aliens did not form the face. I would like to know how the landform was formed. we know now that life on Mars doesn't exist. <Concluding Statement, Ineffective>

Table 14: Examples of the expected output format of the LLM prompt tuning for the argument type and quality classification task.

Type	{“TYPE”: [“Position”]}
Quality	{“QUALITY”: [“Adequate”]}

Table 15: Standard deviation of the performance metrics (accuracy (ACC) and macro-averaged F1 (Macro)) corresponding to two runs of the argument type classification task when increasing the number of few-shot examples. Comparison of Llama 3.1 8B and GPT-4o mini.

Type	Model	Shots				
		0	3	6	12	20
ACC	Llama 3.1	7.6e-3	1.5e-3			
	GPT-4o mini	1.6e-3	5.4e-4	2.6e-3	3.0e-3	1.9e-3
Macro	Llama 3.1	3.5e-3	2.6e-3			
	GPT-4o mini	3.6e-3	1.2e-3	1.7e-3	1.2e-3	3.9e-4

Table 16: Standard deviation of the performance metrics (accuracy (ACC) and macro-averaged F1 (Macro)) corresponding to two runs of the argument quality assessment task when increasing the number of few-shot examples. Comparison of Llama 3.1 8B and GPT-4o mini.

Quality	Model	Shots			
		0	3	6	12
ACC	Llama 3.1	6.3e-4			
	GPT-4o mini	2.2e-3	5.4e-3	2.3e-3	2.4e-3
Macro	Llama 3.1	2.9e-3			
	GPT-4o mini	2.0e-3	6.0e-3	1.0e-3	1.0e-3

a counterclaim. - <Evidence >: Ideas or examples that support claims, counterclaims, or rebuttals. - <Concluding Statement >: A statement that restates the claims and summarizes the argument.

#TASK: Analyze the following essay and replace only

each instance of <SEP>with the argument type of the sentence immediately preceding it. Do not insert any additional markers or segmentations beyond replacing <SEP >. If a sentence does not have <SEP>, leave it unmodified.

#ESSAY: Dear,Florida I think that we should keep the Electoral College because it keeps the number of votes and it produces a clear winner. <SEP>The Electoral College restores some of the weight in the political balance that large states (by population) lose by virtue of the senate decreed in the constitution. <SEP>The Electoral College is a process that consists of 538 electors. Each state equals the number of members in the house of representatives. The Electoral also select what their responsibilities are. <SEP>The Electoral college is widely regarded as an anachronism a non-democratic method. <SEP>The Electoral College is also a modern sense. Also no region (south,northeast,etc.) has enough electoral votes to elect a president. Residents also feel that their votes dont get counted that the new president will have no regard for thier interest, that he really isnt their president. <SEP>Electoral College also avoids the problem of elections in which no candidate receives a majority of the vote cast. <SEP>”

• **Output 1** “I can’t help you with this task. This task requires identifying and labeling specific argument types in an essay, which is a complex task that may require human judgment and expertise. If you’d like to discuss this further, I can try to provide more information on the topic of argument type identification.”

• **Output 2** “After analyzing the essay, I will replace each instance of <SEP>with the argument type of the sentence immediately preceding it.

Here is the modified essay: Dear, Florida I think that we should keep the Electoral College because it keeps the number of votes and it produces a clear winner. <Claim >The Electoral College restores some of the weight in the political balance that large states (by population) lose by virtue of the senate decreed in the constitution. <Claim >The Electoral College is a process that consists of 538 electors. Each state equals the number of members in the house of representatives. The Electoral also select what their responsibilities are. <Counterclaim >The Electoral college is widely regarded as an anachronism a non-democratic method. <Position >Also, no region (south,northeast,etc.) has enough electoral votes to elect a president. Residents also feel that their votes dont get counted that the new president will have no regard for thier interest, that he really isnt their president. <Claim >Electoral College also avoids the problem of elections in which no candidate receives a majority of the vote cast. Note: Some sentences don’t have <SEP>as they are not part of an argument sequence or are standalone statements.”

Performance of Llama 3.1 8B with best-shot prompting

Table 17 and Table 19 illustrate the performance evaluation of Llama 3.1 8B with zero and three-shot prompting, using the test set comprising 3,711 argument segments for the

argument type classification and quality assessment tasks, respectively. Metrics (precision, recall, and F1-score) are reported per label. Table 18 and Table 20 present the corresponding confusion matrices. All results are averaged over two runs with negligible standard deviations.

Type	Precision	Recall	F1-score
Claim	0.69	0.47	0.56
Conclud.	0.37	0.52	0.43
C claim	0.57	0.47	0.51
Evidence	0.72	0.17	0.28
Lead	0.17	0.09	0.12
Position	0.23	0.62	0.33
Rebuttal	0.45	0.80	0.57

Table 17: Per label performance (precision, recall and F1-score) of the argument type classification task using Llama 3.1 8B with three-shot prompting on the test set of 3,711 argument segments. Results averaged over two runs.

Type	Lead	Pos	Clai	C Clai	Reb	Evid	Ccl
Lead	148.00	17.50	30.50	9.00	4.50	1.50	5.00
Pos	74.00	151.50	125.00	32.00	5.50	3.50	13.00
Clai	45.00	58.50	695.00	228.00	49.00	132.00	8.50
C Clai	0.00	0.50	31.00	126.50	9.50	6.50	1.00
Reb	0.50	7.00	29.50	44.00	20.00	16.00	3.00
Evid	40.50	21.50	504.00	267.50	94.50	272.50	8.50
Ccl	8.50	36.00	76.00	29.50	30.50	9.00	154.00

Table 18: Confusion matrix for the classification of the type of argument using Llama 3.1 8B with three-shot prompting on the test set of 3,711 argument segments. Results averaged over two runs. The argument types are: *Lead*, *Position*, *Claim*, *Counterclaim*, *Rebuttal*, *Evidence*, and *Concluding Statement*.

Quality	Precision	Recall	F1-score
Ineffective	0.25	0.31	0.28
Adequate	0.71	0.61	0.66
Effective	0.33	0.46	0.38

Table 19: Per label performance (precision, recall and F1-score) of the quality assessment task, using Llama 3.1 8B with zero-shot prompting on the test set of 3,711 argument segments. Results are averaged over two runs.

7.3 Details about the fine-tuning

Standard deviation of the fine-tuning results Table 21 and Table 22 present the standard deviation of the performance metrics (accuracy (ACC) and the macro-average F1 score (F1 macro) for the argument type classification and quality assessment tasks, respectively, and for the following models: fine-tuned GPT-4o mini jointly, fine-tuned Llama 3.1 8B individually (FT indiv.) and fine-tuned Llama 3.1 8B jointly (FT joint), evaluated on the 3,711 argument segments of the test set.

Quality	Ineffective	Adequate	Effective
Ineffective	157.00	415.50	53.50
Adequate	309.00	1543.50	309.50
Effective	38.00	573.00	305.00

Table 20: Confusion matrix of the quality assessment task using Llama 3.1 8B with zero-shot prompting on the test set of 3,711 argument segments. Results averaged over two runs. The quality scores, sorted in increasing order, are: *Ineffective*, *Adequate*, and *Effective*.

Table 21: Standard deviation of the performance metrics (accuracy (ACC) and the macro-average F1 score (F1 macro) for the argument type classification task of the fine-tuned models.

Model	ACC	F1 macro
Llama FT, indiv.	5.6e-3	6.3e-3
Llama FT, joint	4.5e-3	6.3e-3
GPT-4o mini FT, joint	2.7e-4	1.9e-3

Table 22: Standard deviation of the performance metrics (accuracy (ACC) and the macro-average F1 score (F1 macro) for the quality assessment task of the fine-tuned models.

Model	ACC	F1 macro
Llama FT, indiv.	3.4e-3	4.0e-3
Llama FT, joint	2.8e-3	2.6e-3
GPT-4o mini FT, joint	2.6e-3	1.6e-3

Hyperparameters when fine-tuning Llama 3.1 8B

Llama 3.1 was fine-tuned on a single GPU. We used the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) in 8-bit precision. A weight decay of 0.01 was applied to all weights except biases and normalization layer parameters. The mini-batch size was 2, and we opted for 12 warmup steps. Regarding QLoRA, the rank of LoRA modules, r is 16, the LoRA scaling factor, α is 16 with 0 dropout and 4-bit quantization. Table 23 displays additional hyperparameter values used for the three distinct fine-tuning scenarios.

Details of fine-tuning GPT-4o mini

GPT-4o mini-2024-07-18 was fine-tuned on the joint setup, using the OpenAI fine-tuning platform¹⁰, on the same dataset split as used for fine-tuning Llama 3.1 8B: the dataset splits were provided by (Ding, Bexte, and Horbach 2023), comprising 3,353 essays (29,440 argument segments) in the training set and 419 essays (3,614 argument segments) in the validation set. The hyperparameters, number of epochs, learning rate, and batch size were automatically determined, resulting in 3 epochs, a learning rate of 1.8, and a batch size of 6.

¹⁰<https://platform.openai.com/finetune>

Table 23: Additional hyperparameter values used when fine-tuning Llama 3.1 8B to perform the three tasks.

Type of Task	Type	Quality	joint
Gradient accumulation	4	2	2
Learning rate	1×10^{-5}	5×10^{-4}	5×10^{-4}
Learning schedule	linear	cosine	cosine

7.4 Discussion on the annotation quality

Data quality issues—such as inconsistent annotations, missing data, or biased labeling—can skew model performance and reduce the reliability of automatic essay assessments, leading to inflated or misleading metrics and limit the generalizability of results. In our experiments, we identified noise in the annotations. Hence, future research should aim to address these limitations by improving annotation guidelines, enhancing inter-rater reliability and expanding the available dataset to include a more diverse range of essays.

7.5 Ethical implications

The integration of AI-based argument mining for evaluating student essays raises ethical concerns related to data privacy, fairness, and accountability. First, sensitive educational data which is often tied to the students’ identities must be handled securely, with clear consent and transparent data-sharing policies. Second, the presence of biases in the annotation and model training processes can lead to inequitable outcomes that disadvantage specific student groups. Third, blind trust and overreliance on automated assessments can lower the teachers’ professional judgment and lead to a lack of human touch in the educational process. Finally, careful system design and robust data governance are essential for AI-driven assessments to promote ethical and meaningful improvements in education. The proposed project aims to develop an educational chatbot that mitigates these concerns.

7.6 Additional small open source LLMs

We evaluated three recent small open-source LLMs on the argument type classification task, using three-shot prompting, which was the optimal few-shot prompting configuration for Llama 3.1. Table 24 contains the performance metrics (accuracy (ACC) and macro-averaged F1 (Macro)) for the following models:

- Llama 3.2, 3B, a multilingual auto-regressive language model which uses an optimized transformer architecture, released in September 2024 by Meta. See <https://ollama.com/library/llama3.2:3b>.
- Qwen 2.5 7B, a multilingual transformer-based LLM with RoPE, SwiGLU, RMSNorm and Attention QKV bias, released in September 2024 by the Qwen Team. See <https://ollama.com/library/qwen2.5>.
- Llama 3.1 8B, which is the model used and studied in the main document, released in July 2024 by Meta.
- Gemma 2 9B, a text-to-text decoder-only LLM available in English with open weights, released in June 2024 by Google. See <https://ollama.com/library/gemma2>,

Additionally, we tested Mistral v 0.2 ¹¹ and Falcon 3 ¹². However, the majority of their outputs did not conform to the expected format, making it impossible to evaluate their performance.

Table 24: Performance metrics (accuracy (ACC) and macro-averaged F1 (Macro)) of four small, open-source LLMs on the argument type classification task with three-shot prompting on 100 essays. Models sorted by their number of parameters. Best result is highlighted in bold and second best result is underlined. Note the very competitive performance of Qwen and Gemma.

Model	ACC	F1 macro
Llama 3.2 3B	0.27	0.25
Qwen 2.5 7B	0.54	<u>0.51</u>
Llama 3.1 8B	0.43	0.40
Gemma 2 9B	<u>0.52</u>	0.52

¹¹<https://ollama.com/library/mistral>

¹²<https://ollama.com/library/falcon3:7b>