Leveraging Small LLMs for Argument Mining in Education: Argument Component Identification, Classification, and Assessment

Anonymous Submission

Abstract

Argument mining algorithms analyze the argumentative structure of essays, making them a valuable tool for enhancing education by providing targeted feedback on 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. This paper proposes leveraging open-source, small Large Language Models (LLMs) for argument mining through few-shot prompting and fine-tuning. These models' small size and open-source nature ensure accessibility, privacy, and computational efficiency, enabling schools and educators to adopt and deploy them locally. Specifically, we perform three tasks: segmentation of student essays into arguments, classification of the arguments by type, and assessment of their quality. We empirically evaluate the models on the "Feedback Prize – Predicting Effective Arguments" dataset of grade 6–12 students essays and demonstrate how fine-tuned small LLMs outperform baseline methods in segmenting the essays and determining the argument types while few-shot prompting yields comparable performance to that of the baselines in assessing quality. This work highlights the educational potential of small, open-source LLMs to provide real-time, personalized feedback, enhancing independent learning and writing skills while ensuring low computational cost and privacy.

Keywords

argument mining, educational feedback, small LLMs, accessibility, privacy„ computational efficiency deployment, argument segmentation classification quality assessment

1. Introduction

Writing well-structured essays can be challenging for students, as they require not only quality argument components but also cohesive connections between them [31]. 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 provide[1]. 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 [23, 5].

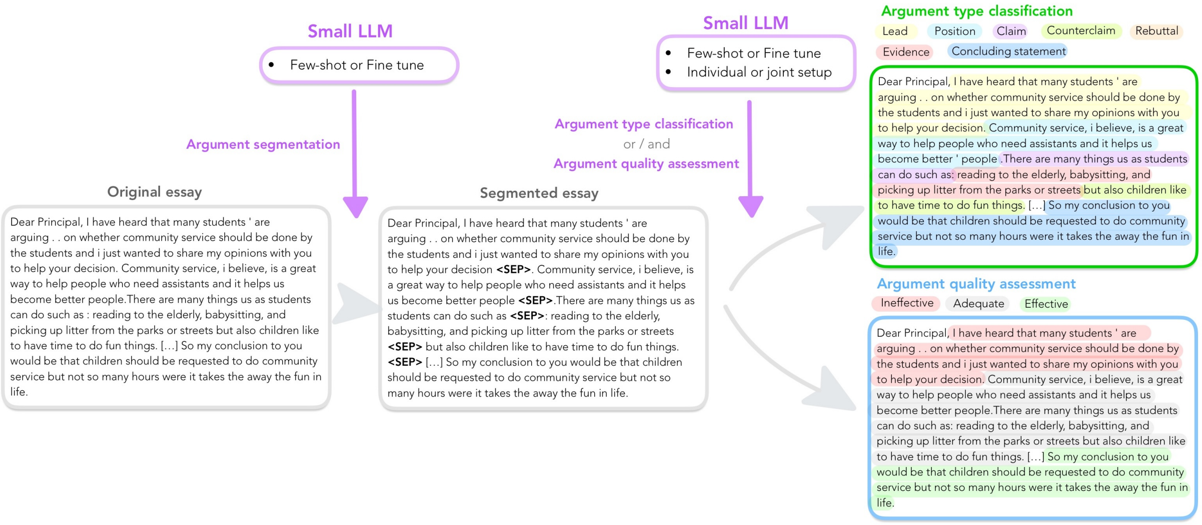
Effective argument mining entails several subtasks, such as segmenting the essay into distinct argument components, classifying their type, assessing their quality and establishing relationships between them [16]. These tasks can be performed sequentially or in parallel. State-of-the-art approaches in argument mining rely on encoder or encoder-decoder deep neural network-based architectures [2]. However, to date, decoder-only models remain underexplored for certain subtasks, suggesting a promising area for future exploration [37].

While high-performing argument mining methods have potential, they are often difficult to access and scale, especially in educational settings [21]. Automated Essay Scoring (AES) systems offer an alternative solution for grading essays, focusing on overall evaluation rather than on assessing individual arguments [15]. However, limited research has explored the integration of argument mining with argument quality assessment. This work aims to bridge this gap by providing detailed feedback that helps students critically analyze their arguments and enhance their writing skills [13].

CMNA'25:The 25th International Workshop on Computational Models of Natural Argument

© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

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



Dear Principal, I have heard that many students ' are arguing . . on whether community service should be done by the students and i just wanted to share my opinions with you to help your decision. Community service, i believe, is a great way to help people who need assistants and it helps us become better people.There are many things us as students can do such as : reading to the elderly, babysitting, and picking up litter from the parks or streets but also children like to have time to do fun things. [...] So my conclusion to you would be that children should be requested to do community service but not so many hours were it takes the away the fun in life.

Dear Principal, I have heard that many students ' are arguing . . on whether community service should be done by the students and i just wanted to share my opinions with you to help your decision **<SEP>**. Community service, i believe, is a great way to help people who need assistants and it helps us become better people **<SEP>**.There are many things us as students can do such as **<SEP>**: reading to the elderly, babysitting, and picking up litter from the parks or streets **<SEP>** but also children like to have time to do fun things. **<SEP>** [...] So my conclusion to you would be that children should be requested to do community service but not so many hours were it takes the away the fun in life.

Figure 1: Overview of the proposed framework. Given an essay as input, the objective is to first segment it into arguments, then classify the argument types, and assess their quality using small open source LLMs —Qwen 2.5 7B, Llama 3.1 8B, and Gemma 2 9B. These tasks are performed either individually or jointly through two learning approaches: few-shot prompting or fine-tuning.

We propose leveraging open-source[2], small Large Language Models (LLMs) to perform a complete pipeline in argument mining (from segmenting the text in arguments to classifying their type and assessing their quality) in an educational context. By means of few-shot prompting and fine-tuning, these models can execute these tasks locally on the student's laptop, ensuring accessibility and maintaining computational efficiency. Figure 1 depicts the proposed 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 methodology. Section 4 presents and analyses our experimental results.

Finally, Section 5 provides a discussion of the findings, followed by a conclusion and an outline of the limitations in Section 6.

2. Related work

Argument mining Argument mining is a complex field that aims to identify, classify, and analyze argumentative structures within text [23], drawing inspiration from frameworks, such as Toulmin's model of argumentation [35]. Argument mining involves numerous subtasks [2], including argument detection, classification, assessment, and relation prediction, making end-to-end solutions particularly challenging [5]. Thus, despite its importance, few studies tackle the full argument mining pipeline due to its complexity and methodological diversity. [6, 3, 27].

State-of-the-art methods in argument mining typically rely on deep neural networks [2]. Recently, advancements in Large Language Models (LLMs) have pushed the field forward. For instance, T5 [29] has been applied effectively to argument mining tasks [22], while models like Longformer [13] and BERT-based approaches [21] have demonstrated competitive performance across various subtasks. More complex systems provide end-to-end solutions by combining models like BART [24] with prompting and graph-based approaches [33], or by leveraging graph prefix tuning to enhance discourse-level understanding [32]. Recently, Gorur et al. [17] demonstrates that prompt-tuned, open-source models like Llama-2 [36] and Mixtral [20] can outperform state-of-the-art RoBERTa-based baselines [30] in identifying agreement and disagreement relations among arguments. However, to the best of our

---

[2] We use the term open-source to refer to LLMs that are freely available with at least open-weights.

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.

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 [21]. Beyond the tasks of argument segmentation and classification, assessing the quality of arguments is essential for evaluating their persuasiveness and coherence [37]. Existing approaches, such as Longformer-based classification methods [12] and graph-based frameworks [26], 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 [5]. Moreover, incorporating discourse-level features has been shown to improve performance by offering deeper insights into argument structures [9].

Contributions 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 segmentation, 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 and models

Given a human-written essay, the goal is to perform three tasks: (1) segmenting it into arguments; (2) classifying each argument into one of seven categories—*Lead, Position, Claim, Counterclaim, Rebuttal, Evidence, Concluding Statement*; and (3) assessing the quality of each argument using three levels —*Ineffective, Adequate, Effective.* Segmentation is conducted first, followed by argument type classification and quality assessment. These latter tasks can be performed independently or jointly.

We investigate the effectiveness of three open-source, small LLMs, namely Qwen 2.5 7B [38], Llama 3.1 8B [14] and Gemma 2 9B [34], for these tasks using both few-shot prompting and fine-tuning. We compare against a state-of-the-art baseline [12, 13] and GPT-4o mini [1] to shed light on the performance of open-source vs closed-source models. Appendix A.7 provides additional details on these models, as well as experimental results for three additional small, open-source LLMs (Llama 3.2 2B, DeepSeek R1 7B, and OLMo 2 7B).

3.2. Few-shot prompting

The first approach applies few-shot prompt-tuning sequentially: first, the LLM is prompted to segment the essay into arguments. Then another prompt is used for argument type classification and/or quality assessment. The prompts are built by concatenating the following textual elements:

1. Few-shot examples are given in the format depicted by the examples included in Tables 7, 8 and 9 in Appendix A.3, depending on the task. The label of the type of argument or its quality is added between each argument in the essay. The example essays are extracted from the training set split used in Ding et al. [13] and described in Section 3.3.

2. The essay. The segmentation task takes the original essay as input. For argument type classification and quality assessment, the essay is provided in a pre-segmented format, as illustrated in Table 7 in

Appendix A.3. In this format, each argument is separated by a designated *SEP* label, which corresponds to the output of the segmentation task.

3. The query, specifying the role of the LLM and the overall instructions for the task. The detailed formulations for the segmentation, argument type classification and quality assessment tasks are provided in Table 5 in Appendix A.2.2.

4. Output requirements. For the argument type classification and argument quality assessment tasks, the model is asked to generate the output in a specific JSON format, described in the prompt as follows: *"For the given* argument component, *identify its [type] and/or [quality]. Provide the output as a JSON object with the key: [TYPE] and/or [QUALITY]."* For segmentation, the output is described in the prompt as follow: "*Place <SEP> immediately at the end of each segment. Preserve all original words, spacing, and order.*"

5. The specific argument of the essay that the LLM is asked to analyze in the argument type classification and argument quality assessment tasks.

Examples of the expected output format and the handling of incorrect outputs are detailed in Appendix A.4. After segmentation, classifying the entire text at once may cause label mismatches, misaligning them with arguments. To simplify this task for smaller LLMs, we classify one argument per prompt. However, the full essay is still provided as context, given that the type classification and quality assessment of each argument 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 it 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 arguments. However, this approach did not deliver good performance with the small 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. For output examples, see Appendix A.3.

3.3. Fine-tuning

The second approach involves fine-tuning an LLM for the following tasks: (1) argument segmentation; (2) argument type classification; (3) argument quality assessment; and (4) argument type classification and quality assessment together. The three small, open-source models were fine-tuned on Google Colab using an A100 GPU on each of the tasks. To establish a comparative upper bound, we also fine-tuned GPT-4o mini using OpenAI's API. Further details on the fine-tuning of this model can be found in Appendix A.2.3.

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 [7, 8]. The dataset encompasses the seven argument types derived from Toulmin's argumentation model [35]: *Lead, Position, Claim, Counterclaim, Rebuttal, Evidence,* and *Concluding Statement*; a quality assessment for each argument is also provided: *Ineffective*, *Adequate*, or *Effective*.

For fine-tuning, we used the split provided by Ding et al. [13] consisting of 3,353 essays (*i.e*, 29,440 arguments) for the training set and 419 essays (*i.e.*, 3,614 arguments) for the validation set. Detailed statistics of this dataset split can be found in Ding et al. [12, 13].

---

[3] https://www.kaggle.com/competitions/feedback-prize-effectiveness/data

Setting We performed the fine-tuning of the small, open-source LLMs using the *SFTTrainer* module from the *TRL* library[4]. We employed *Unsloth*[5] to optimize performance and reduce memory usage.

Additionally, we incorporated Low-Rank Adaptation (LoRA) [19] and Quantized Low-Rank Adaptation (QLoRA) [10] 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 Appendix A.2.3. Regarding GPT 4o-mini, we used OpenAI's API for fine-tuning this model.

Input sequence and target Both the input sequence and the fine-tuning target were formatted in accordance with the previously described few-shot prompting methodology. For the segmentation task, the input was the original essay, while the target output was the corresponding essay segmented by a designated *SEP* label (See Table 7, in Appendix). In contrast, for the joint task of argument type classification and quality assessment, the input consisted of the essay already segmented with the *SEP* labels (See Table 7, in Appendix). The target, in this instance, was defined as the same essay further partitioned into discrete arguments, with each argument interleaved with labels specifying both the type of argument and its quality (See Table 10, in Appendix). This approach ensures methodological consistency across tasks and facilitates a systematic evaluation of model performance on both segmentation and combined argument type classification and quality assessment tasks. See Appendix A.2.3 for more details.

Inference The fine-tuned models were run on an Apple M1 Pro laptop with 32 GB RAM using Ollama[6], an open-source framework that enables users to run, create, and share LLMs locally on their machines. We did not provide any few-shot examples demonstrating how to perform the task or how to specify the output format. To evaluate performance, we used the same test set as Ding et al. [13]. 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 non-fine-tuned models. See Table 3 in Appendix, for a summary of experiment variants.

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 by Ding et al. [13], consisting of 419 essays with a total of 3,711 arguments. Detailed statistics of this dataset split can be found in Ding et al. [12, 13]. For the segmentation task, the essay has to be segmented into arguments. For the argument type classification task, the segmented arguments need to be classified into one of seven types: *Lead, Position, Claim, Counterclaim, Rebuttal, Evidence,* and *Concluding Statement.* In the case of the quality assessment task, the possible values are: *Ineffective, Adequate, Effective.* See Table 5 in the Appendix A.2.2 for a description of each label.

4.2. Performance metrics

To assess the efficiency of the proposed methods, we report the following metrics that consider the imbalance in the distribution of labels.

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 per-label basis.

---

[4] https://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

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.

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 [8]. This metric is a standard in the argument mining community [3, 33, 13, 27].

Segmentation F1 score We compute the F1 score for segmentation at the token level using the BIO framework, where each token at the *beginning* of an argument is tagged as *B*, tokens *inside* the argument are tagged as *I* and *O* denotes when a token is not part of an argument, which is not the case in our task as all the tokens are supposed to belong to an argument.

Type and quality F1 score We adopt the evaluation method used by Ding et al. [13]. A predicted argument (S_p) with at least 50% of overlap with a gold argument (S_g) is considered a *match* i.e: $\min(o_{gold}, o_{pred}) > 0.5$, where: $o_{gold} = \frac{|S_g| \cap |S_p|}{|S_g|}$ is the overlap of the predicted argument with the gold argument and $o_{pred} = \frac{|S_g| \cap |S_p|}{|S_p|}$ is the overlap of the gold argument with the predicted argument. Matched predicted arguments are considered *true positives* if they are of the same type (or quality) as the gold argument. Otherwise, they are classified as a false negative. Unmatched predicted arguments are considered false positives and labeled as *Echec*.

Spelling errors LLMs are required to reproduce each input essay—originally written by children—segmented with the *SEP* separator. Because these essays contain numerous spelling errors, smaller LLMs often attempt to correct them automatically, thereby altering the text and complicating direct comparisons with the ground truth. Since our goal is not to address spelling errors and the dataset labels do not account for them, we first correct the essays before providing them to the LLM to ensure accurate and consistent segmentation. To do so we use `language-tool-python`, a wrapper for LanguageTool[9].

4.3. Baselines

We compare our method with several encoder-based variants following the approach proposed by Ding et al. [13]. Their framework employs BERT [11] for argument type and quality assessment and a Longformer [4] for token-level segmentation, leveraging the Longformer's ability to handle long-text classification (see Appendix A.2.1). Notably, their method is the only approach in the literature that utilizes this dataset for the same tasks while processing segmented essays as input. Additionally, as mentioned earlier, we compare our results with GPT-4o mini [1], both vanilla and fine-tuned (using the same fine-tuning data as with the other small LLMs, see Appendix A.2.3) versions, to have a sense of the upper performance bound achieved by a commercial model.

4.4. Results

4.4.1. Argument segmentation

Figure 2 compares the performance of the small, open-source models on the segmentation task, evaluated in their best few-shot settings (either zero or three) and fine-tuned configurations. Additionally, we include the Longformer as a state-of-the-art baseline, and GPT-4o mini (with three-shot learning and fine-tuned) as a commercial upper bound. Error bars indicate the standard deviation across three runs. All LLM models surpass the Longformer's performance, demonstrating the advantages of large-scale pretraining and transfer learning. Fine-tuned models exhibit substantial performance gains over their few-shot counterparts, underscoring the effectiveness of supervised adaptation. Among all small open-source models, the fine-tuned Llama 3.1 8B achieves the highest F1 score of **87.52**, an increase of **18.05** points or **26.00%** in segmentation performance over the baseline (Longformer) and **6.41** points but

---

[8] See the formula in Appendix A.5.1

[9] https://languagetool.org

3.7% below GPT-4o mini's performance. Figure 4, in Appendix, presents the overlap of the inferred segmentation with the gold segmentation across models. Fine-tuned Llama 3.1 8B achieves the highest overlap with the ground truth, outperforming Longformer and aligning with the macro F1 results. Figure 5 in the Appendix A.5.2 reports the average number of arguments per essay across models for further segmentation analysis.

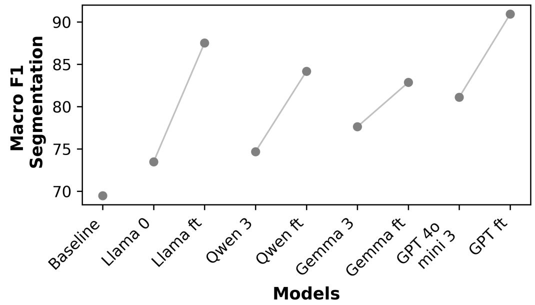


Figure 2: Macro-averaged F1 scores [%] for the argument segmentation task across models. Comparison of small open-source models (Qwen 2.5 7B, Llama 3.1 8B, Gemma 2 9B) in the best few-shot (zero or three-shot) and fine-tuned (ft) settings with the baseline (Longformer) and GPT-4o mini, both with three-shot and fine-tuned. Error bars depict the standard deviation.

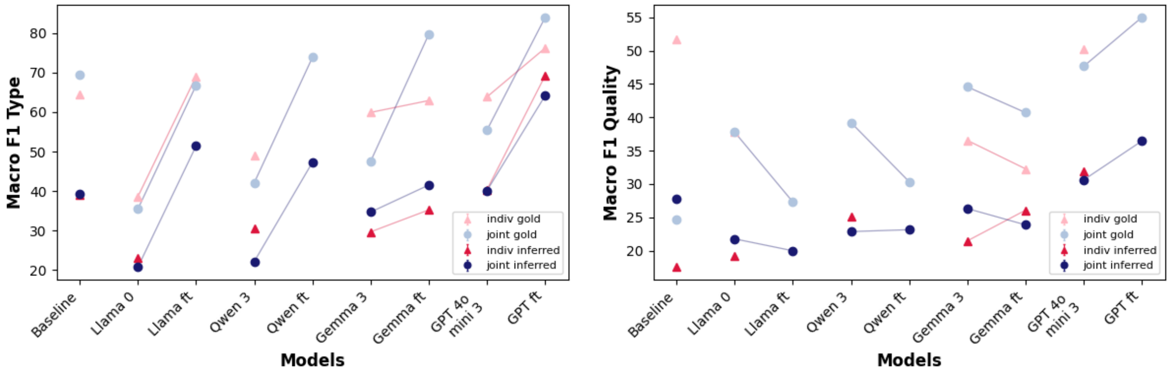


Figure 3: Macro-averaged F1 scores [%] for the argument type classification (left) and quality assessment (right) across models. Comparison of three small open-source models (Qwen 2.5 7B, Llama 3.1 8B, Gemma 2 9B) in the best few-shot (zero or three-shot) and fine-tuned (ft) settings with the baseline and GPT-4o mini (few-shot and fine-tuned). The results highlighted in transparent colors correspond to the evaluation with the gold segmentation whereas the darker colors correspond to inferred segmentation. In the case of gold segmentation, the baseline corresponds to a BERT model with two prediction heads. In the case of inferred segmentation, the segmentation is carried out by a Longformer followed by a classification with BERT. Circles represent the joint setup (both type and quality classification are performed at the same time) whereas triangles correspond to the individual setup (type and quality classification are performed separately). Error bars show the standard deviation.

4.4.2. Argument type and quality assessment

Figure 3 shows the macro-averaged F1 scores for the argument type (left) and quality (right) tasks, respectively, across various models and experimental settings. The figure compares the performance of the three small open-source models in the best few-shot (either zero or three) and fine-tuned settings, along with the baseline and GPT 4o mini (three-shot and fine-tuned). The F1 scores are reported for two task setups: *individual* classification (indiv.) and *joint* classification of both argument type and quality together (joint). Results are shown for two segmentation conditions: *gold* (provided) and *inferred*. Transparent colors indicate the gold condition, while less transparent colors represent inferred segmentation. Circles and triangles correspond to the joint and individual setups, respectively. Error bars represent the standard deviation of the results.

Argument type classification Regarding the type classification task, models using the gold segmentation consistently outperform the models with inferred segmentation and the baseline. We also observe significant improvements when fine-tuning the small, open-source LLMs, both with the gold and inferred segmentation. The best-performing model with the gold segmentation is Gemma 2 9B fine-tuned, with an F1 score of **79.74**, which is **10.27** points or **14.78%** larger than the baseline (BERT) but **5.1%** below GPT-4o mini's performance. The best-performing model with inferred segmentation is Llama 3.1 8B fine-tuned, with an F1 score of **51.45**, which is **12.27** points or **31.32%** larger than the baseline (Longformer + BERT) but **19.85%** lower than GPT-4o mini fine-tuned. Generally, models yield better performance in the joint than in the individual setup.

Argument quality assessment Regarding the argument quality assessment task, models using the gold segmentation also consistently outperform the models with inferred segmentation. In this case, the best-performing small open-source model is Gemma 2 9B with three-shot learning, achieving an F1 score of **44.56**, which is **19.96** points or **81.14%** larger than the baseline's performance but **18.98%** lower than the GPT-4o mini fine-tuned performance. With inferred segmentation, the best performing small, open-source LLM (Gemma 2 9B with three-shot learning) achieves similar performance to the baseline, both of them below that of GPT-4o mini. Interestingly, fine-tuning, in this case, tends to worsen the performance across models, particularly when given the gold segmentation.

Note that the results for Llama 3.1 Qwen 2.5 and GPT-4o mini fine-tuned are absent in the individual setup due to repeated inference failures. Refer to Section 6 for more details.

4.4.3. Label-level performance analysis

To better understand the previously reported performance figures, we summarize the results at the label level for the best-performing model (Llama 3.1 8B fine-tuned in the argument type classification task and Gemma 2 9B three-shot learning in the argument quality assessment task) in the joint set up and the inferred segmentation configuration. Tables 1, 2, (and 12 in Appendix) depict the precision, recall and F1 score for the argument type classification, quality assesment and segmentation tasks, respectively. Furthermore, Tables 14, 15 and 13 in Appendix contain the corresponding confusion matrices.

Type	Precision	Recall	F1-score
Lead	48.48	96.97	64.58
Position	52.63	58.94	55.59
Claim	41.17	69.86	51.75
C claim	48.34	53.70	50.71
Rebuttal	51.67	47.09	47.49
Evidence	54.91	86.33	67.11
Concluding	77.92	71.28	74.36

Table 1
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 inferred argument segmentation.

Argument type classification As seen in Table 1 the model is the most accurate in classifying *Concluding statements*, followed by *Evidence*. In contrast, the model exhibits the lowest performance when classifying *Rebuttals* and *Counterclaims*. Interestingly, the model has very high recall (96.97%) but low precision (48.48%) when classifying *Lead* arguments, suggesting that it often over-labels arguments as *Lead*. There is large misclassification rates of *Claim*, *Position*, and *Evidence*, into *Echec* entries, confirming that segmentation errors contribute to lowering the performance in the classification task, see Table 14 in the Appendix.

Quality assessment Regarding quality assessment, the best performance of the model is in the *Adequate* label, struggling with the *Ineffective* (low recall) and especially the *Effective* (low precision) labels. In fact, *Adequate* tends to be over-predicted by the model. Part of the classification errors are also due to segmentation mistakes, as reflected by the *Echec* column in the confusion table in Table 15 in the Appendix.

Quality	Precision	Recall	F1-score
Ineffective	42.00	29.78	34.84
Adequate	35.98	62.74	45.70
Effective	15.86	54.34	24.65

Table 2
Performance evaluation per label in the argument quality assessment task using Gemma 2 9B three-shot model on the joint setup and inferred segmentation.

5. Discussion

In this paper, we have explored the potential of three small, open-source LLMs—namely, Qwen 2.5 7B, Llama 3.1 8B, Gemma 2 9B—to perform three argument mining tasks in an educational setting: argument segmentation, argument type classification, and argument quality assessment. We have experimented with both few-show prompting and fine-tuning, comparing small open-source LLMs to commercial LLMs (GPT-4o mini) and state-of-the-art encoders. From these extensive experiments, we draw several findings.

First, small and open-source LLMs are able to effectively perform argument mining tasks with significantly better performance than state-of-the-art baselines (Longformer and BERT). Commercial small LLMs, such as GPT4o-mini, yield the best performance.

Second, different approaches provide the best results depending on the task. Whereas fine-tuned models consistently outperform few-shot prompting in the argument segmentation and type classification tasks, we observe the opposite behavior in the argument quality assessment task.

Third, 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 [25]. This finding is particularly evident in the case of Llama 3.1 8B, which exhibits the best performance in tasks with zero-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.

Fourth, joint fine-tuning setups where both tasks were carried out at the same time tend to yield better results than individual setups where the tasks were performed independently, showing a strong link between classifying argument types and assessing their quality [8].

Fifth, the automatic segmentation and 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 [10] [37].

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 [21].

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 for argument segmentation and type classification such that small,

---

[10] See Appendix A.6 for a discussion on the annotation quality

open-source LLMs significantely outperform state-of-the-art approaches by 18.05 points in argument segmentation and 10.27 points in argument type classification. Our experiments also illustrate the value of joint setups for improved argument type classification. By focusing on models running locally on students' personal computers, our research promotes accessibility and resource efficiency, illustrating the potential of open-source, small LLMs as a promising tool for educational applications.

Limitations

Our work is not exempt from limitations that we plan to address in future work. First, the performance on the argument quality assessment task is low for all models, including state-of-the-art methods. We hypothesize that the poor performance might be due to the quality of the annotated data, as suggested by other authors [7, 8]. Hence, we plan to improve the annotation quality and create a more reliable dataset to be shared with the research community.

Second, the fine-tuned Qwen 2.5 7B, Llama 3.1 8B, and GPT4o mini in the individual setup were unable to perform the argument type classification and quality assessment tasks due to repeated failures during inference. While the exact cause remains to be investigated, potential reasons include the hyperparameters used for the fine-tuning not being suitable for the individual setup or instability in the fine-tuned models.

Finally, we evaluated small LLMs using the *PERSUADE 2.0 corpus* dataset, which consists solely of English high-school writing. As a result, it remains uncertain whether findings can be generalized to other educational contexts and languages, which could be explored in future work.

Ethical considerations

The integration of AI-based argument mining for automatically 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 over-reliance on automated assessments can lower the teachers' professional judgment and lead to a lack of human touch in the educational process. The proposed approach is part of a larger project aiming to develop an educational chatbot that mitigates these concerns by leveraging small, open-source LLMs that run locally on the students' computers and by consciously measuring and mitigating biases both in the training data and the models.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

[2] Sakshi Arora, Ajay Rana, and Archana Singh. 2023. Argument mining: A categorical review. In *Modern Electronics Devices and Communication Systems: Select Proceedings of MEDCOM 2021*, pages 353–367. Springer.

[3] Jianzhu Bao, Yuhang He, Yang Sun, Bin Liang, Jiachen Du, Bing Qin, Min Yang, and Ruifeng Xu. 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*, pages 10437–10449.

[4] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

[5] Elena Cabrio and Serena Villata. 2018. Five years of argument mining: A data-driven analysis. In *IJCAI*, volume 18, pages 5427–5433.

[6] Lang Cao. 2023. Autoam: An end-to-end neural model for automatic and universal argument mining. In *International Conference on Advanced Data Mining and Applications*, pages 517–531. Springer.

[7] 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 (persuade) corpus 1.0. *Assessing Writing*, 54:100667.

[8] Scott Andrew Crossley, Perpetual Baffour, Yu Tian, Alex Franklin, Meg Benner, and Ulrich Boser. 2023. A large-scale corpus for assessing written argumentation: Persuade 2.0. *Available at SSRN 4795747*.

[9] Darshan Deshpande, Zhivar Sourati, Filip Ilievski, and Fred Morstatter. 2023. Contextualizing argument quality assessment with relevant knowledge. *arXiv preprint arXiv:2305.12280*.

[10] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. QLoRA: Efficient finetuning of quantized LLMs. *Advances in Neural Information Processing Systems*, 36.

[11] Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[12] Yuning Ding, Marie Bexte, and Andrea Horbach. 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)*, pages 124–133.

[13] Yuning Ding, Marie Bexte, and Andrea Horbach. 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*, pages 13052–13063.

[14] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

[15] Rujun Gao, Hillary E Merzdorf, Saira Anwar, M Cynthia Hipwell, and Arun Srinivasa. 2024. Automatic assessment of text-based responses in post-secondary education: A systematic review. *Computers and Education: Artificial Intelligence*, page 100206.

[16] Luke Gessler, Shabnam Behzad, Yang Janet Liu, Siyao Peng, Yilun Zhu, and Amir Zeldes. 2021. Discodisco at the disrpt2021 shared task: A system for discourse segmentation, classification, and connective detection. *arXiv preprint arXiv:2109.09777*.

[17] Deniz Gorur, Antonio Rago, and Francesca Toni. 2024. Can large language models perform relation-based argument mining? *arXiv preprint arXiv:2402.11243*.

[18] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

[19] 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*.

[20] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

[21] Omid Kashefi, Sophia Chan, and Swapna Somasundaran. 2023. Argument detection in student essays under resource constraints. In *Proceedings of the 10th Workshop on Argument Mining*, pages 64–75.

[22] Masayuki Kawarada, Tsutomu Hirao, Wataru Uchida, and Masaaki Nagata. 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)*, pages 2002–2014.

[23] John Lawrence and Chris Reed. 2020. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

[24] Mike Lewis. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

[25] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and

Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

[26] Santiago Marro, Elena Cabrio, and Serena Villata. 2022. Graph embeddings for argumentation quality assessment. In *EMNLP 2022-Conference on Empirical Methods in Natural Language Processing*.

[27] Gaku Morio, Hiroaki Ozaki, Terufumi Morishita, and Kohsuke Yanai. 2022. End-to-end argument mining with cross-corpora multi-task learning. *Transactions of the Association for Computational Linguistics*, 10:639–658.

[28] Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2024. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*.

[29] 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. *Journal of machine learning research*, 21(140):1–67.

[30] Ramon Ruiz-Dolz, Jose Alemany, Stella M Heras Barberá, and Ana García-Fornes. 2021. Transformer-based models for automatic identification of argument relations: A cross-domain evaluation. *IEEE Intelligent Systems*, 36(6):62–70.

[31] Marlene Scardamalia and Carl Bereiter. 1987. Knowledge telling and knowledge transforming in written composition. *Advances in applied psycholinguistics*, 2:142–175.

[32] Yang Sun, Guanrong Chen, Caihua Yang, Jianzhu Bao, Bin Liang, Xi Zeng, Min Yang, and Ruifeng Xu. 2024. Discourse structure-aware prefix for generation-based end-to-end argumentation mining. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11597–11613.

[33] Yang Sun, Muyi Wang, Jianzhu Bao, Bin Liang, Xiaoyan Zhao, Caihua Yang, Min Yang, and Ruifeng Xu. 2024. Pita: Prompting task interaction for argumentation mining. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5036–5049.

[34] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

[35] Stephen E Toulmin. 2003. *The uses of argument*. Cambridge university press.

[36] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

[37] Henning Wachsmuth, Gabriella Lapesa, Elena Cabrio, Anne Lauscher, Joonsuk Park, Eva Maria Vecchi, Serena Villata, and Timon Ziegenbein. 2024. Argument quality assessment in the age of instruction-following large language models. *arXiv preprint arXiv:2403.16084*.

[38] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

A. Appendix

A.1. Detailed experimental setup and methodology

The experimental setup consists of multiple configurations based on different task types, segmentation methods, adaptation strategies, and model choices. Table 3 provides an overview of the experiment variants. The tasks include argument segmentation, argument type classification, and argument quality assessment. Two segmentation approaches are considered: gold segmentation (ground truth) and inferred segmentation (automatically performed by the model). The setup can be either individual, where argument type and quality are classified separately, or joint, where both are assessed together. We explore two adaptation strategies: few-shot learning (ranging from zero to four-shot) and fine-tuning.

The models used in the experiments fall into three categories: encoder-based models, small open-source LLMs, and a proprietary model, GPT-4o mini.

Category	**Options**
Task	- Argument segmentation - Argument type classification - Argument quality assessment
Segmentation	- Gold - Inferred
Setup	- Individual: type or quality - Joint: type and quality
Adaptation strategy	- Few-shot (zero to four-shot) - Fine-tuned
Models	- Encoder-based: - BERT - BERT with two heads - Longformer - Longformer with two heads - Small open source LLMs: - Llama 3.2 3B - OLMo 2 7B - Qwen 2.5 7B - DeepSeek R1 7B - Llama 3.1 8B - Gemma 2 9B - GPT 4-o mini

Table 3
Summary of experiment variants: detailing different tasks, segmentation methods, experimental setups, adaptation strategies, and the models used.

A.2. Further details on adaptation strategies

A.2.1. Encoder-based baseline

The following encoders are used for different task variants, following the approach proposed by Ding et al. [13]:

- **BERT** [11] for argument type classification and argument quality assessment (individual setup) with segmentation given.
- **BERT with two prediction heads** [13] for joint argument type classification and argument quality assessment (joint setup) with segmentation given.
- **Longformer** [4] for segmentation via token classification.
- **Longformer with two prediction heads** [13] for segmentation and argument type classification and segmentation and argument quality assessment (joint setup).

The training was conducted for 10 epochs and using the same specific setting used in Ding et al. [13]. The evaluation follows the same methodology used to assess the LLMs' performance.

A.2.2. Few-shot learning

Few-shot prompting queries Complementing the Section 3.2 in the main paper, Table 4 presents the expected output format for different tasks, while Table 5 lists the queries used to prompt the LLMs.

Tasks	Few-shot output format
Type	{"TYPE": ["Position"]}
Quality	{"QUALITY": ["Adequate"]}
Type and Quality	{"TYPE AND QUALITY": ["Position", "Adequate"]}

Table 4
Expected output format for few-shot learning for the argument type classification and quality assessment task in the individual and joint setups.

Task	Instructions
Segmentation	#TASK: Segment the following essay into distinct argument components. After each argument component, insert the marker <SEP>. Keep the original text in the same order without adding, removing, or altering any words (other than inserting the <SEP> markers). #GUIDELINES: Identify each coherent segment that forms a logical unit of the argument (e.g., claims, premises, evidence, or conclusions).
Type	You are a strict AI evaluator specializing in detecting the type of argument components 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 a counterclaim. - *Evidence*: Ideas or examples that support claims, counterclaims, or rebuttals. - *Concluding Statement*: A statement that restates the claims and summarizes the argument.
Quality	You are a strict AI evaluator specializing in assessing the quality of argument components in essays. Each component should be rated as one of the following: - *Ineffective*: The component is unclear, unconvincing, or poorly structured. - *Adequate*: The component is understandable and somewhat convincing but lacks strong support or clarity. - *Effective*: The component is well-structured, clear, and strongly supports the argument.

Table 5
Queries used in LLM few-shot prompting (Section 3.2) for segmentation, argument type classification, and quality assessment tasks.

A.2.3. Fine-tuning

Following the dataset split used in [13], we fine-tuned the small LLMs and GPT-4o-mini on a training set of 3,353 essays (29,440 argument segments) and evaluated them on a validation set of 419 essays (3,614 argument segments).

Input sequences and target formats Complementing the Section 3.3 in the main paper, Table 6 provides the input sequences and target formats used during the fine-tuning for the different tasks. See Tables 7, 8, 9 and 10 in Appendix A.3 for examples of these formats.

Fine-tuning small, open source LLMs Qwen 2.5 7B, Llama 3.1 8B, and Gemma 2 9B were 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. Gradient accumulation is set to 4, with a learning rate of $1e-4$ or $5e-4$, depending on the model and setup configuration, and a cosine

Tasks	Input Format	Target Format
Segmentation	Essay	Essay + *SEP* separator
Type (Indiv.)	Essay + *SEP*	Essay + *TYPE* separators
Quality (Indiv.)	Essay + *SEP*	Essay + *QUALITY* separators
Type + Quality	Essay + *SEP*	Essay + *TYPE* and *QUALITY* sep.

Table 6
Input sequences and target formats for different tasks when fine-tuning the models.

learning rate schedule. Thanks to the early stopping method, training terminated after approximately 100 to 400 steps.

Fine-tuning GPT-4o mini GPT-4o mini-2024-07-18 was fine-tuned, using the OpenAI fine-tuning platform[11]. 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.

A.3. Prompting and handling output format

A.3.1. Examples of prompt and output 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 descovered yet **<SEP>** [...] Though people were not satified about how the landform was a natural landform, in all, we new that alieans 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>**

Table 7
An example of essay segmentation format used as few-shot and fine-tuned output format for the segmentation task and input format 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 descovered yet **<Claim>** [...] Though people were not satified about how the landform was a natural landform, in all, we new that alieans 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 8
An example of the few-shot and fine-tuned output format used for the argument type classification task.

Table 7 presents an example of essay segmentation format used as few-shot and fine-tuned output format for the segmentation task and input format for the argument type classification task. Table 8, Table 9 and Table 10 present examples of the 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 they directly reflect the source text.

---

[11] https://platform.openai.com/finetune

> 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 descovered yet **<Adequate>** [...] Though people were not satified about how the landform was a natural landform, in all, we new that alieans 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>**

Table 9
An example of the few-shot and fine-tuned output format used for the argument 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. **<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 descovered yet **<Claim, Adequate>** [...] Though people were not satified about how the landform was a natural landform, in all, we new that alieans 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 10
An example of the few-shot and fine-tuned output format used for the argument type classification and quality assessment tasks.

A.4. Incorrect outputs format during LLM inference

Handling incorrect outputs During inference, the LLM's responses did not always adhere to the specified output format required to perform properly the argument mining tasks. In such cases, the inference was repeated up to five times. If the issue persisted, the argument (or the essay) was discarded.

Examples of incorrect outputs Table 11 presents two examples of incorrect outputs generated with Llama 3.1 8B for the argument type task, along with their corresponding prompt. 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 the essay is missing: "The Electoral College is also a modern sense", and Llama 3.1 8B has omitted the classifications of some arguments.

Fequency of incorrect output format across models and experiments Among all the tested LLMs, fine-tuned models required more inference retries and experienced higher failure rates compared to their few-shot counterparts. Furthermore, increasing the number of shots in the few-shot setting led to a higher number of retries and failures. Notably, Qwen 2.5 7B in the few-shot setting exhibited the highest failure rate among all models.

A.5. Further analyses

A.5.1. Details about the macro-averaged F1 score

The macro-averaged F1 score is the mean of the F1 scores for each label, treating all labels equally. Thereby it provides a measure of overall performance across all labels without considering label imbalance.

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:

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

Prompt“#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 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 becuase 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 representives. 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 mordern 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 alson 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 representives. 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 alson 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.”

Table 11

Two examples of incorrect outputs generated with Llama 3.1 8B for the argument type task, along with their corresponding prompt.

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

A.5.2. Segmentation analysis

Table 12 depicts the precision, recall, and F1 score for the argument segmentation task, and Table 13 the corresponding confusion matrix.

Segm.	**Precision**	**Recall**	**F1-score**
B	66.01	88.32	75.55
I	99.79	99.18	99.48

Table 12

Performance evaluation (precision, recall and F1-score per label) for the segmentation task using the fine-tuned Llama 3.1 8B model.

Llama 3.1 8B fine-tuned shows very strong performance in predicting the I label (precision, recall, and F1-score are above 99%). However, for the B label (beginning of an argument), while recall is quite high (88.32%), precision is noticeably lower (66.01%), suggesting the model sometimes over-predicts

Segm.	B	I
B	254.33	131.00
I	33.67	15881.00

Table 13
Confusion matrix for the segmentation task using the fine-tuned Llama 3.1 8B model on the joint setup.

beginnings. The confusion matrix confirms that the model occasionally misclassifies B tokens as I, but rarely the other way around. Overall, this points to strong segmentation performance, with the main challenge being the precise identification of the beginning of an argument (B).

Figure 4 shows the overlap with the gold segmentation and predicted segmentation across models.

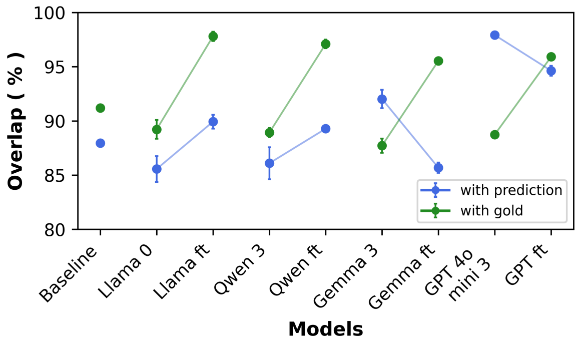


Figure 4: Overlap, in %, with the gold segmentation and predicted segmentation across models. Comparison of small open-source models (Llama 3.1 8B, Qwen 2.5 7B, Gemma 2 9B) in the few-shot and fine-tuned (ft) settings with the baseline (Longformer) and GPT-4o mini few-shot and fine-tuned for the joint setup. Error bars correspond to the standard deviation.

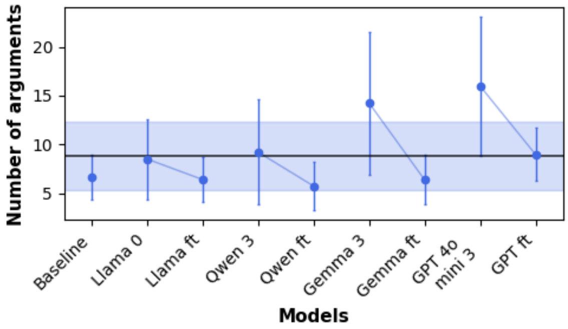


Figure 5: Average number of arguments Comparison of small open-source models (Qwen 2.5 7B,Llama 3.1 8B, and Gemma 2 9B) in few-shot and fine-tuned (ft) settings with the baseline (Longformer) and GPT 4o min few-shot and fine-tuned for the joint setup. Error bars show the standard deviation.

A.5.3. Argument type classification analysis

As a complementary analysis of Table 1 in the main paper, Table 14 displays the confusion matrix using the fine-tuned Llama 3.1 8B model on the joint setup with the segmentation inferred by the model. Note that the entire *Echec* row consists of zeros as this label is not present in the ground truth dataset, and it is counted when the predicted argument does not match the ground truth argument.

A.5.4. Quality assessment analysis

As a complementary analysis of Table 2 in the main paper, Table 15 displays the confusion matrix of fine-tuned Gemma 2 9B on the joint setup with the segmentation inferred by the model.

Type	Lead	Pos	Clai	C Clai	Reb	Evid	Ccl	Ech
Lead	10.33	8.67	0.33	0.00	0.00	1.00	0.00	1.00
Pos	0.00	23.00	4.33	0.67	0.67	0.67	3.33	11.67
Clai	0.00	5.33	51.33	3.67	1.67	7.67	1.33	53.67
C Clai	0.00	0.33	1.00	9.67	0.00	0.33	0.00	8.67
Reb	0.00	0.33	0.33	0.00	4.67	0.33	0.33	3.33
Evid	0.33	1.33	16.33	3.67	2.67	74.00	5.00	31.33
Ccl	0.00	0.00	0.00	0.33	1.00	1.67	24.67	4.00
Ech	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 14
Confusion matrix for the classification of the type of argument using the fine-tuned Llama 3.1 8B model on the joint setup without the gold segmentation. The argument types are: *Lead, Position, Claim, Counterclaim, Rebuttal, Evidence,* and *Concluding Statement. Echec* is accounted when the predicted argument doesn't match with the gold segment.

Quality	Ineffective	Adequate	Effective	Echec
Ineffective	32.33	16.00	2.00	26.67
Adequate	72.67	95.22	10.67	86.33
Effective	4.00	40.33	15.00	34.67
Echec	0.00	0.00	0.00	0.00

Table 15
Confusion matrix for the quality assessment task using the fine-tuned Gemma 2 three-shot model on the joint setup without the gold segmentation. The quality scores, sorted in increasing order, are: *Ineffective*, *Adequate*, and *Effective*. *Echec* is accounted when the predicted segment doesn't match with the gold segment.

A.6. 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 generalization 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.

A.7. Additional small open source LLMs

We evaluated three recent small open-source LLMs on the argument type classification task, using three-shot prompting. Table 16 contains the macro-averaged F1 score for the following models:

- **Llama 3.2, 3B.** Llama 3.2 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.
- **OLMo 2 7B.** OLMo 2 is the latest iteration of the fully open language model, featuring dense autoregressive models with enhanced architecture and training methodologies, released in November 2024 by the Allen Institute for AI [28]
- **Qwen 2.5 7B**. Qwen 2.5 is a multilingual transformer-based LLM with RoPE, SwiGLU, RMSNorm and Attention QKV bias, released in September 2024 by the Qwen Team. [38].
- **DeepSeek R1 7B**. DeepSeek R1 is an open-source large language model designed to enhance reasoning capabilities through reinforcement learning. It rivals other advanced models in tasks such as mathematics, coding, and logical reasoning. Released in January, 2025 by the Chinese AI startup DeepSeek[18].
- **Llama 3.1 8B**, Llama 3.1 is a multilingual large language model optimized for dialogue applications. It supports eight languages and offers a context window of up to 128,000 tokens, enabling it to handle extensive conversational contexts. Released in July 2024 by Meta [14].
- **Gemma 2 9B**, Gemma is a text-to-text decoder-only LLM available in English with open weights, released in June 2024 by Google, [34].

Additionally, we tested Mistral v 0.2 [12] and Falcon 3 [13]. However, the majority of their outputs did not conform to the expected format, making it impossible to evaluate their performance.

Model	**Type**	**Quality**
Llama 3.2 3B	27.56	34.67
OLMo 2 7B	34.28	31.91
Qwen 2.5 7B	<u>42.15</u>	<u>39.20</u>
DeepSeek R1 7B	29.19	37.81
Llama 3.1 8B	38.25	39.11
Gemma 2 9B	**47.55**	**44.56**

Table 16
Macro-averaged F1 of three additional small, open-source LLMs on the argument type classification task with three-shot prompting. Models are sorted by their number of parameters. Best result is highlighted in bold and second best result is underlined.

---

[12] https://ollama.com/library/mistral
[13] https://ollama.com/library/falcon3:7b