End-to-End Argument Mining in Student Essays: A Comparison of Pipeline and Multi-Task Generative Models*

Fahad M. Alzaidee, Tommy Yuan and Peter Nightingale

University of York, Heslington, York YO10 5DD, United Kingdom

Abstract

End-to-end argument mining aims to automatically identify argumentative components and their relationships within unstructured text. This paper investigates the effectiveness of generative large language models (LLMs) for this task on student essays from the Argument-annotated Essays Corpus (AAEC). We compare two strategies: a Pipeline approach, which separates component and relation identification into sequential stages, and a Multi-Task Learning (MTL) framework, which jointly models all subtasks. Both approaches employ the TANL tagging format to represent argumentative components and their relations. Experiments with LLaMA3 and Mistral models show that the Pipeline consistently outperforms MTL, demonstrating its advantage in structuring argumentative discourse.

Keywords

Natural Language Processing, Argument Mining, LLM, Student essays

1. Introduction

Argumentation is a fundamental skill that empowers individuals to present and defend claims through evidence and reasoning. For students, mastering this skill is essential for developing critical thinking and effective communication. Argument Mining (AM), a specialized area within Natural Language Processing (NLP), seeks to automate the identification and structuring of arguments in text, approximating human-like reasoning [1]. By breaking down arguments into subtasks—such as identifying argumentative spans, classifying components, and determining relationships (typically support or attack)—AM constructs structured argumentative frameworks [2].

In recent years, argument mining has gained significant traction across diverse domains. From analyzing scientific literature [3] and enhancing information retrieval systems [4] to enabling automated essay scoring [5] and advancing dialogue analysis [6], its applications are both broad and impactful. As the field continues to evolve, the development of algorithms capable of recognizing and analyzing argument structures with human-like precision remains a central goal, driving innovation in intelligent systems and beyond.

The Argument-annotated Essays Corpus (AAEC), introduced by Stab and Gurevych [7], is a widely used dataset for evaluating argument mining approaches. It models student essays as hierarchical trees of claims and premises. End-to-end argument mining (AM) aims to automatically transform unstructured argumentative text into a structured representation by identifying both the argument components (e.g., claims, premises) and the relationships between them. To address the complexity of this task, two methodological paradigms have emerged. The Pipeline approach decomposes AM into sequential subtasks, training specialized models for each step [7, 8]. Conversely, Multi-Task Learning (MTL) employs a unified neural architecture to jointly model all AM subtasks, enabling shared learning across tasks.learning across tasks.

Recent advancements in Large Language Models (LLMs), such as LLaMA [9], Mistral [10], and Falcon [11], have revolutionized natural language processing. These models, which are based on Transformer frameworks, employ self-attention mechanisms to efficiently capture contextual relationships between

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

fmfa501@york.ac.uk (F. M. Alzaidee); tommy.yuan@york.ac.uk (T. Yuan); peter.nightingale@york.ac.uk (P. Nightingale)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-02-09

tokens. They excel in text comprehension, generation, and adaptability across self-supervised, multi-task, and few-shot learning settings.

End-to-end argument mining (AM) is a challenging task due to the non-linear and complex discourse structure of argumentative texts, as well as the scarcity of annotated datasets. These challenges necessitate innovative approaches to effectively identify argument components and their relationships. In this study, we investigate the efficacy of generative large language models (LLMs) for end-to-end AM by addressing the following research questions:

1. How effective are decoder-only architectures of LLMs in addressing the subtasks of argument mining?
2. To what extent can pipeline and multi-task learning approaches enhance the performance of pre-trained LLMs for argument mining, and which strategy is more effective for capturing discourse structure?

To address these questions, we apply and compare the following strategies for argument mining:

- **Pipeline Approach**: We divide the subtasks of argument mining into two consecutive steps. The first step (component identification) involves segmentation detection and component classification, while the second step (relation identification) focuses on relation detection and classification.
- **MTL with TANL**: We implement a unified framework using Translation between Augmented Natural Languages (TANL) [12], which reframes structured prediction as a text-to-text task. This approach has achieved state-of-the-art results in semantic parsing and is adapted here for argument mining.

2. Related Work

2.1. Argument Mining

The process of argument mining involves several key subtasks:

- **Text Segmentation**: Identifying the boundaries of argumentative segments in a given text [13, 14].
- **Component Classification**: Classifying argumentative segments into categories such as claims and premises [15, 16].
- **Relation Detection**: Identifying and classifying relationships between propositions, such as support or attack [17, 18].

While most works focus on individual subtasks, relatively few address the end-to-end AM problem, which aims to construct argumentation structures directly from unstructured text. Two common approaches are used for end-to-end argument mining:

Pipeline Approach: This approach decomposes the AM task into sequential subtasks. For instance, Persing and Vincent proposed a pipeline method consisting of two stages: first identifying argument components and then classifying the argumentative relations between them. Their approach used Integer Linear Programming (ILP) to perform joint inference over the outputs of argument component identification and relation identification classifiers, incorporating global consistency constraints. However, their results were moderate, achieving F1 scores of 47.1 for component identification and classification and 12.9 for relation identification and classification.

Stab and Gurevych [7] adopted a sequence labeling approach at the token level using a Conditional Random Field (CRF) and tow classifiers based on Support Vector Machines (SVM) for component identification task and relation identification task.A joint model based on ILP is used to optimizes the outcomes the two classifier to detect argumentation structures in persuasive essays.

Multi-Task Learning (MTL): This approach mitigates the limitations of the pipeline method, such as error propagation and the rigid constraints of Integer Linear Programming (ILP), by jointly learning multiple subtasks within a unified framework.

Eger et al. [19] proposed two models: BLCC and LSTM-ER. BLCC approaches argument mining as a sequence tagging problem, whereas LSTM-ER integrates tree-structured and sequential LSTM architectures for end-to-end relation extraction. On the AAEC dataset, LSTM-ER achieved an F1 score of 66.21 for component identification, outperforming BLCC's score of 63.23. For relation identification, LSTM-ER achieved an F1 score of 29.59, while BLCC scored 34.82.

Morio et al. [20] proposed a multi-task learning framework built on a biaffine architecture and longformer model. The framework jointly identifies the argument component spans, their types (e.g., Claim/Premises), and the relations between these components along their relation types.They proposed two models: the ST model, trained on a single dataset, and the MT model, trained on multiple auxiliary corpora and fine tuned on the target dataset. On the AAEC dataset, ST model achieved F1 scores of 75.54 for argument component identification and classification and 55.17 for relation identification and classification while MT model achieved comparable scores.

Closely related to our work, there are several studies that eliminate the need for dependency parsing to frame the complex argumentative structure. Bao et al. [21] proposed a novel generative framework that frames the end-to end argument mining as a simple sequence-to-sequence generation task. The framework employs the pre-trained sequence-to-sequence language model (BART) with two mechanisms: a Constrained Pointer Mechanism (CPM), which acts as an auxiliary task during training and a constrained decoding method during inference, guiding the generation to valid output; and a Reconstructed Positional Encoding (RPE) to mitigate the order biases introduced by the autoregressive generation for modeling long-range dependencies. On the AAEC dataset, their approach achieved F1 scores of 75.94 for argument component identification and classification and 50.08 for relation identification and classification.

Similarly, Kawarada et al. [22] redefined argument mining as a structured prediction task using the TANL framework [12]. They utilized FLAN-T5 XXL to generate structured outputs by annotating argumentative spans, components, and relations. Their method achieved state-of-the-art results across benchmark datasets, including the Argument-annotated Essays Corpus (AAEC), AbstRCT, and the Cornell eRulemaking Corpus (CDCP). Specifically, on the AAEC dataset, they reported F1 scores of 80.15 for component identification and 61.12 for relation identification.

2.2. Generative Models

Large Language Models (LLMs) have garnered substantial attention within the artificial intelligence community due to their advanced capabilities in processing, comprehending, and generating human-like text. These models are trained on extensive textual corpora using autoregressive architectures, which generate text by predicting one token at a time based on the preceding sequence. Built on the Transformer framework [23], LLMs utilize self-attention mechanisms to efficiently capture dependencies between tokens, regardless of their position in the input sequence. Tokens, the fundamental units of text, may represent words or subword fragments depending on the tokenization strategy. By iteratively predicting subsequent tokens, LLMs can uncover intricate linguistic patterns.

Several studies have explored the application of generative models in computational argumentation. Chen et al. [24] investigate the capabilities of large language models (LLMs), such as ChatGPT, Flan, and LLaMA2, in identifying and extracting arguments from text, as well as in generating counterarguments. Gilardi et al. [25] highlight the potential of employing ChatGPT as a tool for text-annotation tasks by evaluating its performance across four tasks: assessing content relevance, detecting stances, identifying topics, and recognizing general frames. Their study compares ChatGPT's annotations against those of crowd workers and expert annotators, demonstrating its competitive accuracy and efficiency. Mirzakhmedova et al. [26] examined the use of large language models (LLMs), specifically GPT-3 and PaLM 2, to assess argument quality. using a zero-shot learning approach. Recent work on Relation-based Argument Mining (RbAM) by Gorur et al. [27] evaluates few-shot prompting for

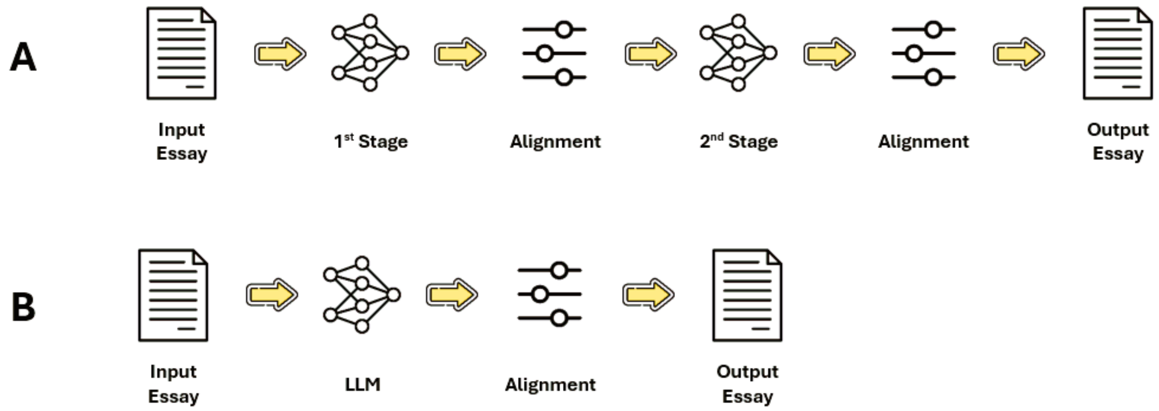


Figure 1: Overview of the pipeline approach (A) and Multi Task Learning (B)

relation-type classification over argument pairs and shows that instruction-tuned LLMs can outperform strong RoBERTa baselines on several benchmarks.

Although prior work demonstrates the potential of generative models in specific argument mining tasks, a unified approach that comprehensively addresses all subtasks of argument mining remains underexplored. Developing such an integrated framework is crucial to advance the robustness and applicability of large language models in computational argumentation.

3. Proposed Method

This study explores end-to-end argument mining using two methodologies: Multi-Task Learning (MTL) and a Pipeline approach (depicted in Figure 1). We employ open-source generative language models—LLaMA 3 (8B), which is proficient in causal language modeling, and LLaMA 3-instruct (8B), an instruction-following fine-tuned variant. Additionally, we use Mistral 7B-Instruct v0.3, a 7 billion parameter pre-trained and fine-tuned model known for its computational efficiency and cost-effective inference. We employ the tagging frameworks proposed by Paolini et al. [12]. Specifically, we use their entity augmentation format to annotate argumentative segments (e.g., Major Claim, Claim, Premise) and their joint entity-relation format to label argumentative relations (Support/Attack). These formats are used to fine-tune the base models. Both the Pipeline and Multi-Task Learning (MTL) approaches use identical inputs and produce the same augmented essay output. In this section, we focus on the Pipeline approach, which follows the traditional two-stage process for argument mining. First, the system detects and categorizes argumentative segments within the raw essay as Major Claims, Claims, or Premises. Then, it evaluates these segments to identify and classify the relationships between them as either supporting or attacking.

3.1. Stage one

Given a raw essay x consisting of n tokens, $x = [x_1, \ldots, x_n]$, the objective of this stage is to perform two argument mining subtasks simultaneously: segment identification and component classification. Each argumentative segment is represented as a 3-tuple (c, s, e), where c denotes the type of the segment (Major Claim, Claim, or Premise), and s and e represent the indices of the first and last tokens of the segment. The output of this stage is an essay with tagged components, following the format used in TANL's entity extraction (Figure 3). To ensure a structured representation, each segment type is assigned a number based on its order of appearance. For example, the first detected claim is labeled Claim 1, the second Claim 2, and so on. This numbering scheme facilitates tracking and referencing argument components in the second stage.

3.2. Stage two

Given the augmented essay from the previous stage, each argumentative segment is identified in the form

$$[\text{segment content} \mid c_n],$$

where the brackets indicate the boundaries of the argumentative segment, and

$$c \in C = \{\text{Major Claim, Claim, Premise}\}.$$

The objective of this stage is to identify argumentative relationships and classify them by type. Each identified relationship is represented as a 3-tuple:

$$(c_n^{\text{src}}, c_m^{\text{tgt}}, r),$$

where c_n^{src} denotes the source argument component, c_m^{tgt} represents the target argument component, and r specifies the relationship type. The indices n and m are defined as

$$n \in \{1, 2, \ldots, N\}, \quad m \in \{1, 2, \ldots, M\}, \quad n \neq m,$$

where N and M represent the total number of argument components of type $c \in C$, ensuring that the source and target components are distinct. The relationship type is drawn from

$$r \in R = \{\text{Support, Attack}\}.$$

To structure the argumentative relationships within the original essay, we adapt the output format of TANL's joint entity and relation extraction task. The encoded format is expressed as

$$[\text{segment content} \mid c_n^{\text{src}} \mid r = c_m^{\text{tgt}}],$$

After predicting the argumentative structure of an essay, we apply the alignment method proposed by Paolini et al. [12]. In the Pipeline, alignment is applied at every stage, while in the MTL approach, it is applied before generating the final output. This method reduces the impact of adding or removing words or phrases between detected argumentative segments, which could otherwise shift their start and end positions.

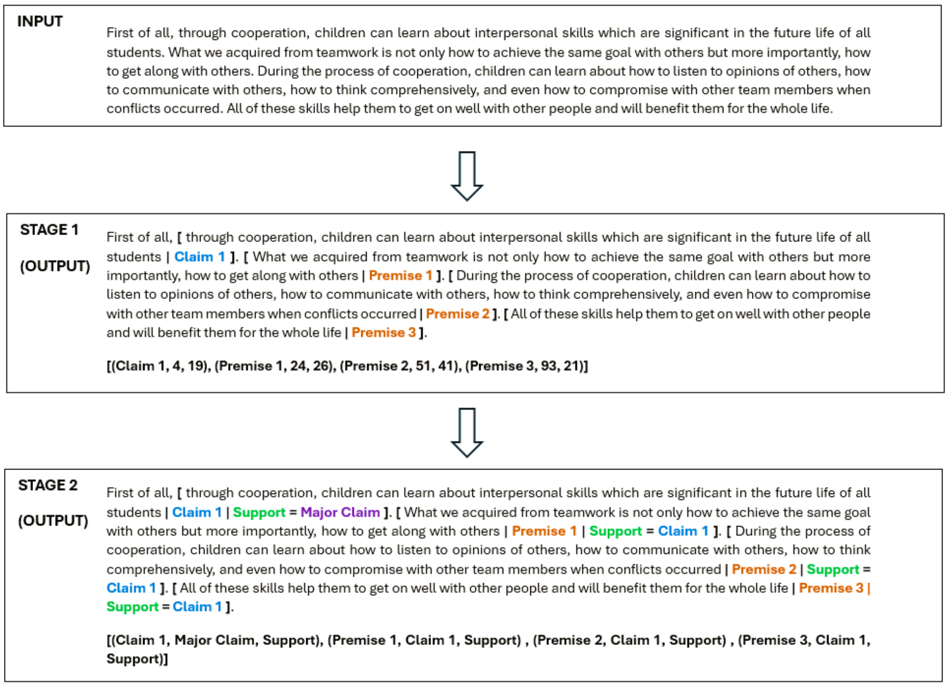
4. Experiments

4.1. Dataset

We evaluate both approaches on the Argument-annotated Essays Corpus (AAEC) to facilitate the study of argumentation structures in persuasive essays. AAEC consists of 402 structured essays on various controversial topics (Table 1 for more details). Of the 402 essays, 322 are set aside for the train set and 80 for the test set. The statistics of the PE dataset are given in Table I. Each essay is annotated with argumentative discourse units (ADUs), which are segments of text that represent individual components of an argument (Major Claim, claim, or premise) and the relationships between ADUs (Support, Attack). The argumentative structure is represented as a hierarchical tree. We fine-tuned and evaluated all models on the essay level rather than on the paragraph level.

4.2. Prompting and Input Preparation for Fine-Tuning

We employ the open-source models LLaMA-3 (8B), LLaMA-3-Instruct (8B), and Mistral-7B-Instruct v0.3 as our base large language models to assess their performance in automating argument mining tasks using both pipeline and MTL approaches. All models are run in an 8-bit quantized configuration, with each weight stored in 8 bits on the GPU. For each approach, we report the average scores from three runs on the test set.



INPUT

First of all, through cooperation, children can learn about interpersonal skills which are significant in the future life of all students. What we acquired from teamwork is not only how to achieve the same goal with others but more importantly, how to get along with others. During the process of cooperation, children can learn about how to listen to opinions of others, how to communicate with others, how to think comprehensively, and even how to compromise with other team members when conflicts occurred. All of these skills help them to get on well with other people and will benefit them for the whole life.

STAGE 1

(OUTPUT)

First of all, **[** through cooperation, children can learn about interpersonal skills which are significant in the future life of all students | **Claim 1** **]**. **[** What we acquired from teamwork is not only how to achieve the same goal with others but more importantly, how to get along with others | **Premise 1** **]**. **[** During the process of cooperation, children can learn about how to listen to opinions of others, how to communicate with others, how to think comprehensively, and even how to compromise with other team members when conflicts occurred | **Premise 2** **]**. **[** All of these skills help them to get on well with other people and will benefit them for the whole life | **Premise 3** **]**.

[(Claim 1, 4, 19), (Premise 1, 24, 26), (Premise 2, 51, 41), (Premise 3, 93, 21)]

STAGE 2

(OUTPUT)

First of all, **[** through cooperation, children can learn about interpersonal skills which are significant in the future life of all students | **Claim 1** | **Support** = **Major Claim** **]**. **[** What we acquired from teamwork is not only how to achieve the same goal with others but more importantly, how to get along with others | **Premise 1** | **Support** = **Claim 1** **]**. **[** During the process of cooperation, children can learn about how to listen to opinions of others, how to communicate with others, how to think comprehensively, and even how to compromise with other team members when conflicts occurred | **Premise 2** | **Support** = **Claim 1** **]**. **[** All of these skills help them to get on well with other people and will benefit them for the whole life | **Premise 3** | **Support** = **Claim 1** **]**.

[(Claim 1, Major Claim, Support), (Premise 1, Claim 1, Support) , (Premise 2, Claim 1, Support) , (Premise 3, Claim 1, Support)]

Figure 2: Example generating augmented essay in the pipeline approach

Table 1
AAEC statistics

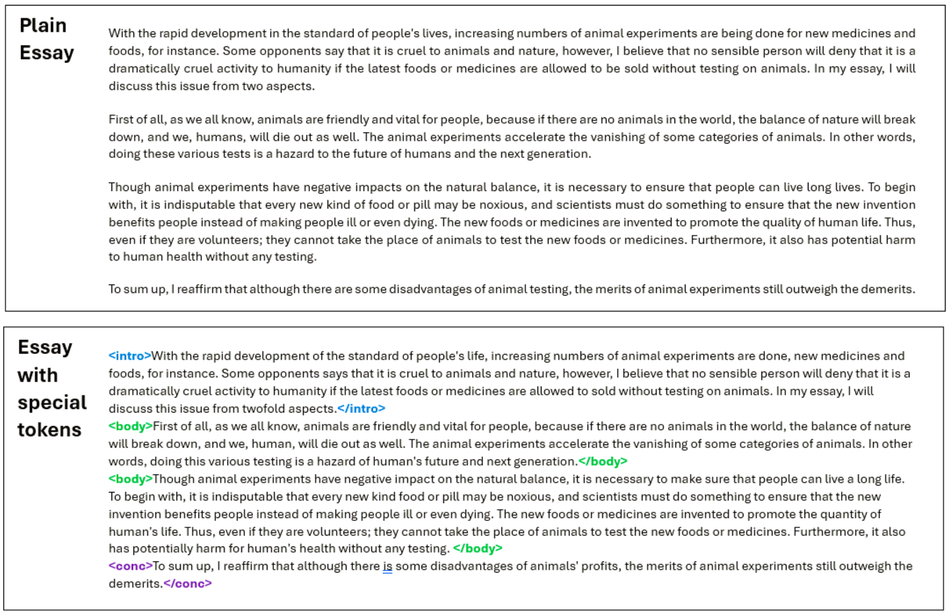
Items	Statistics
Essays	402
Paragraphs	1,833
Sentences	7,116
Tokens	147,271
Major Claims	751
Claims	1,506
Premises	3,832

To guide the models in performing argument mining subtasks, we employ a prompt template adapted from Stanford Alpaca [28], which comprises instruction, input, and output sections (see Appendix A).

We evaluate the model's performance on two types of input: (1) raw essay text and (2) text augmented with special tokens indicating paragraph functions (see Figure 3) The special tokens used are: <**intro**> and </**intro**> to mark the introduction paragraph, <**body**> and </**body**> to mark the body paragraph(s), and <**conc**> and </**conc**> to mark the conclusion paragraph. To address the computational demands of fine-tuning large language models (LLMs), we employ Low-Rank Adaptation (LoRA) [29], a parameter-efficient fine-tuning (PEFT) method. LoRA retains the weights of the original model and integrates trainable low-rank matrices into the transformer layers to simulate weight adjustments. This approximation, based on the principle that the adaptation process has a low "intrinsic rank," reduces the number of trainable parameters to under 1% while maintaining performance levels comparable to full fine-tuning.

4.3. Evaluation Criteria

We assessed the performance of component and relation classification using micro F1 scores, following the approach by Persing and Ng [30]. For component classification, a true positive occurs when a predicted argumentative component matches both the type (e.g., claim, premise) and boundary of a gold standard component. For relation classification, a true positive is counted when the predicted



Plain Essay

With the rapid development in the standard of people's lives, increasing numbers of animal experiments are being done for new medicines and foods, for instance. Some opponents say that it is cruel to animals and nature, however, I believe that no sensible person will deny that it is a dramatically cruel activity to humanity if the latest foods or medicines are allowed to be sold without testing on animals. In my essay, I will discuss this issue from two aspects.

First of all, as we all know, animals are friendly and vital for people, because if there are no animals in the world, the balance of nature will break down, and we, humans, will die out as well. The animal experiments accelerate the vanishing of some categories of animals. In other words, doing these various tests is a hazard to the future of humans and the next generation.

Though animal experiments have negative impacts on the natural balance, it is necessary to ensure that people can live long lives. To begin with, it is indisputable that every new kind of food or pill may be noxious, and scientists must do something to ensure that the new invention benefits people instead of making people ill or even dying. The new foods or medicines are invented to promote the quality of human life. Thus, even if they are volunteers; they cannot take the place of animals to test the new foods or medicines. Furthermore, it also has potential harm to human health without any testing.

To sum up, I reaffirm that although there are some disadvantages of animal testing, the merits of animal experiments still outweigh the demerits.

Essay with special tokens

<intro>With the rapid development of the standard of people's life, increasing numbers of animal experiments are done, new medicines and foods, for instance. Some opponents says that it is cruel to animals and nature, however, I believe that no sensible person will deny that it is a dramatically cruel activity to humanity if the latest foods or medicines are allowed to sold without testing on animals. In my essay, I will discuss this issue from twofold aspects.</intro>
<body>First of all, as we all know, animals are friendly and vital for people, because if there are no animals in the world, the balance of nature will break down, and we, human, will die out as well. The animal experiments accelerate the vanishing of some categories of animals. In other words, doing this various testing is a hazard of human's future and next generation.</body>
<body>Though animal experiments have negative impact on the natural balance, it is necessary to make sure that people can live a long life. To begin with, it is indisputable that every new kind food or pill may be noxious, and scientists must do something to ensure that the new invention benefits people instead of making people ill or even dying. The new foods or medicines are invented to promote the quantity of human's life. Thus, even if they are volunteers; they cannot take the place of animals to test the new foods or medicines. Furthermore, it also has potentially harm for human's health without any testing. </body>
<conc>To sum up, I reaffirm that although there is some disadvantages of animals' profits, the merits of animal experiments still outweigh the demerits.</conc>

Figure 3: raw essay vs essay with added special tokens

source and target components correspond exactly to those in the gold annotations and share the same relation type (e.g., Support, Attack).

4.4. Implementation Details

For this study, both the Multi-Task Learning (MTL) framework and the pipeline approach utilized a unified set of hyperparameters in all base language models (Table 6). Specifically, we applied a batch size of 8, trained for 3 epochs, and enabled early stopping. Optimization was carried out using pagedadamw32bit, with a learning rate of 0.0002, scheduled via a cosine decay strategy, alongside a weight decay of 0.006. Additionally, we employed a warmup ratio of 0.03 to stabilize training. LoRA-specific settings were consistently maintained, with a rank of 32, an alpha value of 64, and a dropout rate of 0.01.

All experiments were conducted on NVIDIA A40 GPU platform using the AdamW optimizer. Training was conducted over 10 epochs with a maximum token length of 1024. Model checkpoints were saved every 10 training steps.

4.5. Evaluation and Results

We conducted an ablation study to assess the impact of adding special tokens that encode paragraph roles. As shown in Table 5, this modification did not improve performance in predicting argumentative component types, indicating that paragraph-level role encoding does not benefit generative models in this task.

Table 4 presents the performance of the pipeline and multi-task learning approaches on the AAEC dataset. The pipeline approach outperforms multi-task learning across all evaluated generative models in both component and relation identification. The LLaMA-3-8B-Instruct model achieves the highest micro F1 scores: 80.02 for component identification and 60 for relation identification.

Table 2 compares our best-performing pipeline model with prior studies that conducted end-to-end argument mining on the AAEC dataset at the essay level. Existing methods include:

- **T2TGen** by Kawarada et al. [22], which frames argument mining as structured prediction using the TANL framework and FLAN-T5 XXL.

- **BLCC and LSTM-ER** by Eger et al. [19], which introduce sequence tagging and tree-structured LSTM architectures for component and relation extraction.
- **MT Model** by Morio et al. [20], employing a biaffine multi-task learning framework built on Longformer for joint modeling of components and relations.
- **BART-CPM** by Bao et al. [21], proposing a generative sequence-to-sequence framework with constrained decoding and positional encoding to model argumentative structure without dependency parsing.

Model	ACC	ARC
BLCC	63.23	34.82
LSTM-ER	66.21	29.56
ST Model	76.55	54.66
T2T-Gen	**80.15**	**61.19**
BART-CPM	75.94	50.08
LLaMA 3 8B (8-bit) (ours)	80.02	60

Table 2
End-to-end argument mining performance on the AAEC dataset at essay level. ACC: Argument Component Classification; ARC: Argument Relation Classification. Bold values indicate the highest micro f1 score in each column.

As shown in Table 2, our model performs competitively with existing approaches. It achieves strong ACC and ARC scores with the LLaMA-3-8B pipeline using 8-bit quantization, demonstrating the efficiency and robustness of our approach.

To evaluate the second stage of our pipeline for **relation classification** on the AAEC dataset, we perform an **oracle analysis** using **annotated essays**. These annotations specify the boundaries and types of argument components in the raw text, effectively removing errors from component identification. **F1 scores** are computed separately for support and attack relations, as well as an overall F1 score. We use **10-fold cross-validation**, testing the model on 10% of the dataset in each fold. **True Positives (TP), False Positives (FP), and False Negatives (FN)** are aggregated across folds to compute dataset-level F1 scores. In this binary setting, a false negative for one class is treated as a false positive for the other.

Table 3 compares our results with those of Gorur et al.[27], who employed an instruction-tuned LLaMA 2 70B (70 billion parameters, 4-bit precision) model for few-shot relation-type classification. In contrast, our best system uses the much smaller LLaMA-3-8B-Instruct (8 billion parameters, 8-bit precision) model yet achieves closely comparable performance across relation types. This clearer comparison highlights that our relation classification framework approaches state-of-the-art few-shot results while being significantly more parameter-efficient.

Model	Support	Attack	Both
LLaMA 2 70B (4-bit)	94	52	**90**
LLaMA 3 8B (8-bit) (ours)	92	49	88

Table 3
F1 scores for support, attack, and overall relation prediction for each model. Bold values indicate the highest performance in the “Both” column.

5. Conclusion and limitation

This study evaluated the Pipeline and Multi-Task Learning (MTL) approaches for end-to-end argument mining using three generative models on the AAEC dataset. The Pipeline approach consistently outperformed MTL in both component and relation classification, highlighting the benefits of task decomposition and sequential processing for decoder-only architectures. While these results are promising, future work should assess the generalizability of the Pipeline framework across diverse datasets and domains. By separating the two main subtasks, the Pipeline design not only improves classification performance but also supports user involvement in refining argumentative structure. This allows users to revise predicted components before relation generation. However, this performance gain comes at the cost of lower inference efficiency. Adapter switching and dual generation steps introduce computational overhead that may limit real-time deployment.

6. Declaration on Generative AI

During the preparation of this work, ChatGPT was employed to assist with grammar and spelling correction as well as paraphrasing. We thoroughly reviewed and edited the text generated with ChatGPT and retain full responsibility for the content presented.

References

[1] R. Mochales Palau, M.-F. Moens, Automatic detection of arguments in legal texts, in: 19th Belgian-Dutch Conference on Artificial Intelligence, Date: 2007/11/05-2007/11/06, Location: Utrecht, The Netherlands, 2007.

[2] J. Lawrence, C. Reed, Argument mining: A survey, Computational Linguistics 45 (2020) 765–818.

[3] Y. Li, W. Chen, Z. Wei, Y. Huang, C. Wang, S. Wang, Q. Zhang, X.-J. Huang, L. Wu, A structure-aware argument encoder for literature discourse analysis, in: Proceedings of the 29th International Conference on Computational Linguistics, 2022, pp. 7093–7098.

[4] C. Stab, T. Miller, I. Gurevych, Cross-topic argument mining from heterogeneous sources using attention-based neural networks, arXiv preprint arXiv:1802.05758 (2018).

[5] Z. Ke, V. Ng, Automated essay scoring: A survey of the state of the art., in: IJCAI, volume 19, 2019, pp. 6300–6308.

[6] S. Saha, S. Das, R. K. Srihari, Dialo-ap: A dependency parsing based argument parser for dialogues, in: Proceedings of the 29th International Conference on Computational Linguistics, 2022, pp. 887–901.

[7] C. Stab, I. Gurevych, Parsing argumentation structures in persuasive essays, Computational Linguistics 43 (2017) 619–659.

[8] V. Niculae, J. Park, C. Cardie, Argument mining with structured svms and rnns, arXiv preprint arXiv:1704.06869 (2017).

[9] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al., The llama 3 herd of models, arXiv preprint arXiv:2407.21783 (2024).

[10] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al., Mistral 7b, arXiv preprint arXiv:2310.06825 (2023).

[11] E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Debbah, É. Goffinet, D. Hesslow, J. Launay, Q. Malartic, et al., The falcon series of open language models, arXiv preprint arXiv:2311.16867 (2023).

[12] G. Paolini, B. Athiwaratkun, J. Krone, J. Ma, A. Achille, R. Anubhai, C. N. d. Santos, B. Xiang, S. Soatto, Structured prediction as translation between augmented natural languages, arXiv preprint arXiv:2101.05779 (2021).

[13] V. W. Feng, G. Hirst, Two-pass discourse segmentation with pairing and global features, arXiv preprint arXiv:1407.8215 (2014).

[14] J. Lawrence, C. Reed, C. Allen, S. McAlister, A. Ravenscroft, Mining arguments from 19th century philosophical texts using topic based modelling, in: Proceedings of the First Workshop on Argumentation Mining, 2014, pp. 79–87.

[15] R. M. Palau, M.-F. Moens, Argumentation mining: the detection, classification and structure of arguments in text, in: Proceedings of the 12th international conference on artificial intelligence and law, 2009, pp. 98–107.

[16] C. Stab, I. Gurevych, Identifying argumentative discourse structures in persuasive essays, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 46–56.

[17] H. Nguyen, D. Litman, Context-aware argumentative relation mining, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 1127–1137.

[18] O. Cocarascu, F. Toni, Identifying attack and support argumentative relations using deep learning, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 1374–1379.

[19] S. Eger, J. Daxenberger, I. Gurevych, Neural end-to-end learning for computational argumentation mining, arXiv preprint arXiv:1704.06104 (2017).

[20] G. Morio, H. Ozaki, T. Morishita, K. Yanai, End-to-end argument mining with cross-corpora multi-task learning, Transactions of the Association for Computational Linguistics 10 (2022) 639–658.

[21] J. Bao, Y. He, Y. Sun, B. Liang, J. Du, B. Qin, M. Yang, R. Xu, 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, 2022, pp. 10437–10449.

[22] M. Kawarada, T. Hirao, W. Uchida, M. Nagata, 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), 2024, pp. 2002–2014.

[23] A. Vaswani, Attention is all you need, Advances in Neural Information Processing Systems (2017).

[24] G. Chen, L. Cheng, L. A. Tuan, L. Bing, Exploring the potential of large language models in computational argumentation, arXiv preprint arXiv:2311.09022 (2023).

[25] F. Gilardi, M. Alizadeh, M. Kubli, Chatgpt outperforms crowd workers for text-annotation tasks, Proceedings of the National Academy of Sciences 120 (2023) e2305016120.

[26] N. Mirzakhmedova, M. Gohsen, C. H. Chang, B. Stein, Are large language models reliable argument quality annotators?, in: Conference on Advances in Robust Argumentation Machines, Springer, 2024, pp. 129–146.

[27] D. Gorur, A. Rago, F. Toni, Can large language models perform relation-based argument mining?, in: Proceedings of the 31st International Conference on Computational Linguistics, 2025, pp. 8518–8534.

[28] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, T. B. Hashimoto, Stanford alpaca: an instruction-following llama model, https://github.com/tatsu-lab/stanford_alpaca (2023).

[29] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, arXiv preprint arXiv:2106.09685 (2021).

[30] I. Persing, V. Ng, End-to-end argumentation mining in student essays, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 1384–1394.

A. Appendix

Model	ACC		ARC	
	MTL	Pipeline	MTL	Pipeline
LLAMA 3 (8B)	77	78.4	54.4	56.3
LLAMA 3-Instruct (8B)	78.2	80.02	54.3	60
Mistral 7B-Instruct v0.3	72.1	73.6	44.1	47.9

Table 4
Compare the predictions made for ACC (Argument Component Classification) and ARC (Argument Relation Classification) from MTL (Multitask learning) and pipeline approaches in terms of micro F1 scores

Model	w/ Special tokens		w/o Special tokens	
	ACC	ARC	ACC	ARC
LLAMA 3 (8B)	77.2	55.3	78.4	56.3
LLAMA 3-Instruct (8B)	79	56.7	80.02	60
Mistral 7B-Instruct v0.3	72.8	46.6	73.6	47.9

Table 5
The effects of adding special tokens to the raw essay

Hyperparameter	Value
Batch size	8
Epochs	3
Early stop	yes
Optimizer	paged_adamw_32bit
Learning rate	0.0002
Learning rate scheduler	cosine
Weight decay	0.006
Warmup ratio	0.03
LoRA rank	32
LoRA alpha	64
LoRA dropout	0.01

Table 6
Hyperparameters for language models

Prompt template for MTL approach

You are an argument analysis expert. Follow these steps **EXACTLY**:

Step 1: Identify Argument Components

- **Major Claim**: The essay’s central argument. Label exactly once or twice if repeated.

- **Claims**: Debatable assertions that support or attack other components. Claims require support from Premises and can be challenged or defended. Label as Claim 1, Claim 2, etc.

- **Premises**: Supporting reasons, evidence, or justification for a Claim or another Premise. Premises do not stand alone as arguments but provide necessary reasoning. They may contain factual evidence, logic-based reasoning, or expert opinions.

Step 2: Determine Argumentative Relationships

- **Support**: The component strengthens or reinforces another.

- **Refute**: The component provides counterarguments that directly challenge another.

- **Contradict**: The component presents opposing ideas without direct refutation.

- Always reference the exact label of the target component (e.g., Major Claim, Claim 1, Premise 2).

- Recognize implicit relationships where direct connections are not explicitly stated but inferred from meaning, tone, and structure.

Formatting Rules (DO NOT DEVIATE):

1. Wrap components in brackets: [Exact Text | Label].

2. For relationships: Add ’| Relationship = Target Component’ after the label.

3. If a component spans multiple sentences, include ALL sentences inside one bracket.

4. Preserve original text order and wording—do NOT add or remove words.

5. Labels must follow this syntax: **Major Claim**, **Claim X**, **Premise X** (X = number).

6. DO NOT include **transitional phrases**, **introductory/reinforcing phrases**, or **stance markers** (e.g., ’I believe,’) inside the brackets.

7. If a sentence contains multiple **distinct** ideas separated by semicolons, conjunctions (’which’, ’but’, etc.), or implicit contrasts, they must be annotated separately. Only annotate the **core argument component** itself.

Additional Guidance:

- Some argumentative components do not function as supporters or refuters, yet they can still be targets of argumentation.

- Transitional words and phrases serve as clues for relationships:

Support indicators: ‘therefore,’ ‘as a result,’ ‘consequently,’ ‘this proves’

Refutation indicators: ‘however,’ ‘on the contrary,’ ‘while,’ ‘although’

Contradiction indicators: ‘despite,’ ‘in contrast,’ ‘alternatively,’ ‘whereas’

- If an argument contains both supportive and opposing elements, label it according to its **dominant** function.

Input:

{essay}

Output:

{Augmented essay}

Table 7

Prompt for annotating arguments in the MTL approach.

Template prompt for the first stage of the pipeline approach
Identify and annotate the argument components in the input, including Major Claims, Claims, and Premises. - **Major Claim**: The essay’s central argument. Label exactly once or twice (if repeated). - **Claims**: Statements that directly assert a position, opinion, or viewpoint regarding the Major Claim. A Claim should make a **debatable assertion** rather than just provide background context. Claims must require support from Premises and can be challenged or defended. - **Premises**: Statements that **provide justification, evidence, or reasoning** for a Claim or another Premise. A Premise **does not stand alone** as an argument but instead supports a Claim. It may contain factual evidence, logic-based reasoning, or expert opinions. **Formatting Rules (DO NOT DEVIATE):** - Wrap components in brackets: [Exact original text | Label]. - If a component spans multiple sentences, include ALL its sentences inside one bracket. - Preserve the original essay’s text and order. Do NOT add/remove words. - Labels MUST follow this syntax: **Major Claim**, **Claim X**, **Premise X** (X = number). - DO NOT include **transitional phrases**, **introductory/reinforcing phrases**, or **stance markers** (e.g., ’I believe,’) inside the brackets. - If a sentence contains multiple **distinct** ideas separated by semicolons, conjunctions (’which’, ’but’, etc.), or implicit contrasts, they must be annotated separately. Only annotate the **core argument component** itself. **The essay is:** {input_text} **Output:** {Augmented essay}

Table 8
Template prompt for the first stage of the pipeline approach.

Template prompt for the second stage of the pipeline approach

You will analyze tagged text spans representing argumentative components.

Task:

1. Identify the argumentative relationships between the tagged text spans.
2. Preserve the original content without modifying or removing any parts.
3. Explicitly state whether each tagged span **SUPPORTS**, **REFUTES**, or **CONTRADICTS** another span.
4. Recognize implicit relationships where direct connections are not clearly stated but can be inferred from meaning, tone, and structure.

Guidelines:

- **SUPPORT**: A span strengthens or reinforces the claim of another component.
- **ATTACK**: A span challenges another component by either:
  - **Refuting** it with direct counterarguments that explicitly disprove or weaken the claim.
  - **Contradicting** it by subtly undermining or presenting an opposing idea without outright refutation.
- Some argumentative components do not function as supporters or refuters, yet they can still be the target of support, contradiction, or refutation.
- Transitional words and phrases serve as clues for relationships. These include:

  Support indicators: *therefore, as a result, consequently, this proves*

  Refutation indicators: *however, on the contrary, while, although*

  Contradiction indicators: *despite, in contrast, alternatively, whereas*
- Consider cases where grammar or phrasing is unconventional—recognize argumentative intent even when structure is not ideal.
- When an argument contains both supportive and opposing elements, identify the **dominant** argumentative function.

Input:

{input_text}

Output:

{Augmented essay}

Table 9

Template prompt for the second stage of the pipeline approach.