On the Role of Key Phrases in Argument Mining

Nilmadhab Das, V. Vijaya Saradhi and **Ashish Anand**
AMaL (Applied Machine Learning) Lab
Department of Computer Science and Engineering
IIT Guwahati, Assam, India, 781039
{nilmadhabdas, saradhi, anand.ashish} @iitg.ac.in

Abstract

Argument mining (AM) focuses on analyzing argumentative structures such as Argument Components (ACs) and Argumentative Relations (ARs). Modeling dependencies between ACs and ARs is challenging due to the complex interactions between ACs. Existing approaches often overlook crucial conceptual links, such as *key phrases* that connect two related ACs, and tend to rely on cartesian product methods to model these dependencies, which can result in class imbalances. To extract key phrases from the AM benchmarks, we employ a prompt-based strategy utilizing an open-source Large Language Model (LLM). Building on this, we propose a unified text-to-text generation framework that leverages Augmented Natural Language (ANL) formatting and integrates the extracted key phrases inside the ANL itself to efficiently solve multiple AM tasks in a joint formulation. Our method sets new State-of-the-Art (SoTA) on three structurally distinct standard AM benchmarks, surpassing baselines by **up to 9.5% F1 score**[1], demonstrating its strong potential.

1 Introduction

Argument mining is a field of study dedicated to the identification and analysis of argumentative structures within a text. It has garnered significant attention recently due to its potential applications in automated essay scoring (Ke et al., 2018), legal decision support (Walker et al., 2018), healthcare applications (Mayer et al., 2020), etc. AM is often divided into four key tasks: *(i) Argument Component Identification (ACI)*, identifying argumentative text spans; *(ii) Argument Component Classification (ACC)*, categorizing these spans into AC types (e.g., claims, premises); *(iii) Argumentative Relation Identification (ARI)*, detecting relationships between the spans; and *(iv) Argumentative Relation Classification (ARC)*, classifying the types of these relationships (e.g., support, attack). Most prior studies have approached these tasks independently using task-specific customization (Morio et al., 2020) or have solved a subset of them either through joint modeling (Bao et al., 2021) or in a pipelined manner (Mayer et al., 2020). Many studies assume that the ACI task has already been completed, focusing solely on other tasks (Liu et al., 2023). Following the existing work, this paper focuses on the joint modeling of ACC and ARI, given the argumentative spans.

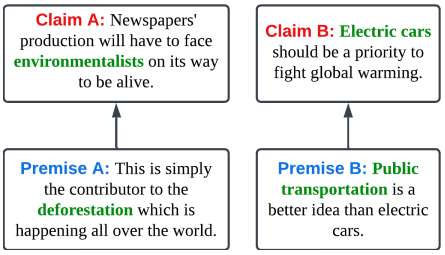

Figure 1: Examples of *Related Key Phrases* between related AC pairs, highlighted in green. Claim A is supported by Premise A, with two key phrases serving as conceptual bridges between these components. Similarly, Premise B attacks Claim B, as they are connected by conceptually opposing key phrases.

Among the various AM tasks, the ARI task is often considered more challenging than the others. It demands the identification of complex interactions between related pairs of ACs. Few studies in the literature have explored these interactions from diverse perspectives. For example, Trautmann (2020) identified *aspect* terms for a deeper understanding of related ACs, whereas Chakrabarty et al. (2021) and Saadat-Yazdi et al. (2023) explored the *Common Sense Knowledge* as explicit features to enrich the representation of the ACs. Recently, Sun et al. (2022) analyzed structural dependencies to identify semantically and syntactically *similar words* between related ACs, which led to performance improvement. While the presence of such similar words can signal the existence of relations between

[1]Our code is available here.

Findings of the Association for Computational Linguistics:
NAACL 2025, pages 5774–5787
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

related AC pairs, they do not fully capture the inherent connections between these ACs. For example, in Figure 1, both Claim B and Premise B contain the phrase *"Electric cars"*, suggesting a potential connection between these ACs. Yet, this similarity does not effectively convey the attacking nature of the relationship between them. Instead, it is the conceptual contrast between *"Electric cars"* in Claim B and *"Public transportation"* in Premise B that reveals the true argumentative nature of the relation, where opposing ideas are at play. This demonstrates the need for a deeper understanding of the relationships between ACs. We refer to such conceptual connections as ***Related Key Phrases***, as they form the crucial conceptual bridge between related pairs of ACs. These phrases serve as crucial indicators, helping to identify ARs. Apart from this, most of the previous studies commonly utilize *Cartesian products* to match all possible pairs of ACs to identify ARs between them (Kuribayashi et al., 2019; Sun et al., 2022). However, these methods are inefficient and often suffer from class imbalance problems, as the majority of AC pairs are unrelated. This imbalance results in sub-optimal performance, highlighting the need for more effective approaches to model those complex relational dependencies in an efficient manner.

The annotation schemes of standard AM corpora vary significantly, with no universally accepted norms regarding the types of ACs and ARs. In some AM datasets, ACs are categorized into coarse-grained classes (Stab and Gurevych, 2017), while others adopt more detailed fine-grained categories (Schaefer et al., 2023). Additionally, the relational structures across corpora differ substantially. For example, certain corpora represent ARs in a tree-structured format (Stab and Gurevych, 2017), whereas others model them as non-tree structures (Niculae et al., 2017). In non-tree AM corpora, if two ARs are given as $(a \rightarrow b)$ and $(b \rightarrow c)$, a transitive relation $(a \rightarrow c)$ is also established, unlike in tree-structured corpora. This introduces additional challenges, particularly for the ARI task. Recently, end-to-end frameworks have emerged to jointly model all four tasks of AM (Morio et al., 2022), aiming to address these complexities of AM corpora by mutual information sharing among different tasks. However, these approaches have shown limited success. This highlights the need for a more focused effort on solving individual or a subset of the AM tasks to address the unique challenges at each task level and improve overall performance.

This paper presents a unified approach to jointly model a subset of the key tasks in AM: ACC and ARI. Our method leverages a text-to-text generation framework, where both input and output are structured using the ***Augmented Natural Language (ANL)*** format. In the input, AC spans are augmented and enclosed with special symbols. On the output side, AC types and their corresponding related spans are embedded with augmented labels, also using special symbols. To invoke the knowledge of *related key phrases*, we first prompt an open-source LLM, guiding it in extracting these phrases between pairs of ACs from the standard AM datasets. Such extracted phrases are then appended to the ANL output sequence, embedding this knowledge directly into the text-to-text generation task. Through extensive experiments across multiple structurally diverse standard AM benchmarks, our proposed approach achieves SoTA results for both tasks, outperforming all existing baselines. These results highlight the strong potential of our method in effectively addressing two distinct AM tasks jointly. In summary:

1. We propose a joint modeling of two key AM tasks, *ACC* and *ARI*, using an *ANL-based* text-to-text generation framework to model both tree and non-tree structured arguments.

2. We integrate the explicit knowledge in terms of *related key phrases* within the target ANL to strengthen its relational representation.

3. We utilize *LLM-based prompting* techniques to extract *related key phrases* between related AC pairs, which in turn enhances the proposed task performance.

4. We conduct extensive experiments achieving SoTA results across datasets and demonstrate strong noise resilience through a *Noise Adaptability Study*. We also assess the *impact of external knowledge* and the *utility of joint modeling* with an *in-depth argumentative structural analysis*.

2 Related Work

2.1 Argument Mining

Most previous studies have tackled different AM tasks either independently (Kuribayashi et al., 2019; Saadat-Yazdi et al., 2023) or sequentially

in a pipelined fashion (Mayer et al., 2020). However, there is a growing trend towards jointly addressing multiple AM tasks within a unified framework (Bao et al., 2021; Morio et al., 2022). In this context, several approaches have been developed to jointly address two pivotal tasks: ACC and ARI. Early works like Stab and Gurevych (2017) introduced joint optimization using Integer Linear Programming (ILP), while Niculae et al. (2017) applied structured SVMs with factor graphs. Later, Potash et al. (2017) proposed a Pointer Network, and Galassi et al. (2018) used residual networks with link-guided training. Further innovations include Kuribayashi et al. (2019) using span representations and LSTMs, and Morio et al. (2020) leveraging task-specific parameterization with biaffine attention. Recent models like Bao et al. (2021) used transition-based approaches with pre-trained language models, while Morio et al. (2022) combined Longformer with biaffine parsing. More recently, Liu et al. (2023) framed AM tasks as machine reading comprehension using BART. However, none explored fine-grained AC interactions, leaving room for improvement in joint modelling.

2.2 Use of External Knowledge in AM

Several studies have emerged for NLP applications that elicit external knowledge by prompting LLMs. For example, Wadhwa et al. (2023) used the LLM (GPT-3) to generate *explanations* about why the related tuples of entities are related. Later, Jiang et al. (2024) also adopted this approach to solve the NER tasks. Specifically in AM, Chakrabarty et al. (2021) used Paragraph-Level Commonsense Transformers (COMET) (Gabriel et al., 2021) as an external knowledge source to solve the implicit premises generation task through a text-to-text generation approach. Similarly, Saadat-Yazdi et al. (2023) utilized a different COMET (Hwang et al., 2020) to solve the relational AM tasks. However, none of the works in AM literature utilizes the knowledge of LLMs to solve the AM tasks.

2.3 Applications of Augmented Natural Language (ANL)

With the rise of generative methods, many NLP tasks are now being approached as ANL-based generation problems. Athiwaratkun et al. (2020) utilized ANL-based generation to solve a range of tasks such as NER, slot labelling and intent classification in a single generation sequence. Zhang et al. (2021) used a similar approach to solve aspect-based sentiment analysis with both extraction style and annotation style ANL sequences. Later, Paolini et al. (2021) introduced TANL, a unified framework for structured prediction tasks by treating them as generative text-to-text translation problems. Liu et al. (2022) framed structures as sequences of actions that build the target step by step. In AM, Kawarada et al. (2024) first applied TANL to jointly handle ACC and ARC tasks using ANL-based target generation. However, the impact of incorporating explicit knowledge into ANL generation remains unexplored in solving AM tasks.

3 Proposed Method

The proposed method consists of two main steps. *First,* the input ANL is represented with special markers to indicate argumentative spans. For example: “Nevertheless, supporters would argue that **[Argument 1]**. They further point out that **[Argument 2]**." Here, the special tokens “[" and “]" denote the start and end of each argument. The output representation further extends this format to include argument classification and relationships: “Nevertheless, supporters would argue that **[Argument 1 | Claim | Argument 2]**". This indicates that **Argument 1** is classified as a **Claim** and is related to **Argument 2**. *Next,* from each argument pair, related key phrases are extracted using `Lamma-3.1-instruct` in a few-shot setup. Then, these extracted key phrases are appended with the target ANL to produce the final ANL output.

This structured input and output ANL are then used to fine-tune a T5 model. During inference, T5 classifies AC spans, predicts argument relations, and generates key phrases as a by-product. Figure 2 illustrates the overall process. Further details are discussed in subsequent sections.

3.1 Key-Phrases Extraction with LLM

Utilizing an open-sourced LLM *Meta-LLaMa-3.1-instruct*, we employ a prompt-based strategy to extract *related key phrases* from the related ACs. To enhance the quality of the extracted key phrases, we adopt a few-shot prompting approach, where we manually create five carefully crafted examples from a standard AM dataset. These examples illustrate both the input and the expected output, helping guide the model towards producing the desired results. To prevent the model from generating irrelevant or random text, we design a well-structured instruction composed of four distinct steps, each

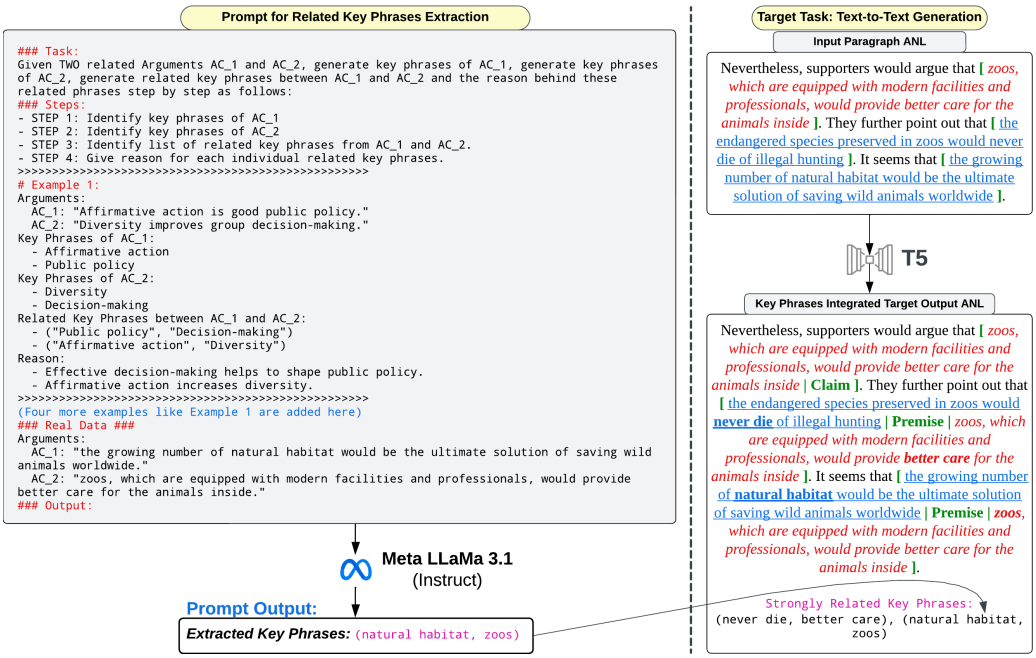

Figure 2: Illustration of the proposed method. On the left, the prompt description for extracting *Related Key Phrases* using the LLM in a 5-shot setting is shown. On the right, the input/output ANL configurations for the proposed generation task are presented. Claim spans are marked in red, Premises are in blue, and augmented labels are in green. The *Related Key Phrases* within the AC spans are in **bold** and are explicitly appended at the end of the output.

aimed at constraining the extraction process (See Figure 2). The first step consists of extracting key phrases from AC_1, focusing on the most relevant phrases that reflect its core argument. In the second step, the model performs a similar extraction from the AC_2. Notably, both AC_1 and AC_2 may contain more than one central point. Next, in the third step, those extracted phrases from both the ACs are matched by identifying conceptual links to establish a connection between these arguments. In the fourth step, the model provides an explanation of how and why the chosen pairs of related key phrases are linked. The reasoning is expressed through meaningful sentences that articulate the logical or thematic connections, ensuring that the extracted links are relevant and coherent. By reasoning this way, the model is forced to reduce the likelihood of extracting unrelated key phrases.

3.2 Joint Formulation of ACC & ARI

We define the proposed ANL formatting based on five key elements: (i) the input paragraph, $W = w_1, w_2, w_3, \ldots, w_n$, where n is the total number of tokens in W. We denote a text span from w_i to w_j in W as $w_{i:j}$; (ii) a list of ACs, $C = \{(c_i, t_i, s_i, e_i)\}_{i=1}^{n}$, where c_i represents the AC span, t_i specifies its type (e.g., Claim, Premise), s_i is the start index, and e_i is the end index of the AC; (iii) a list of ARs, $R = \{(h_j, t_j)\}_{j=1}^{m}$, where $h_j, t_j \in C$ are the head and tail ACs, respectively; (iv) a list of LLM-extracted related key phrases from all the ARs is denoted by $KP = \{(e_{1_h}, e_{1_t}), (e_{2_h}, e_{2_t}), \ldots, (e_{k_h}, e_{k_t})\}$, where e_{i_h} and e_{i_t} denote the related key phrases extracted from related head and tail ACs, respectively; (v) a set of special symbols $S = \{[,], (,), |, =\}$, where "[" marks the start of an augmented label, "]" marks the end of an augmented label, "(" and ")" are used to enclose the phrases, "|" separates different augmented labels, and " = " connects the related AC spans. Now, consider two arbitrary ACs C_p and C_q from the set C, which are related by an AR R_k from R, where $h_k = C_p$ and $t_k = C_q$. The spans of C_p and C_q are denoted by $w_{c_p^s:c_p^e}$ and $w_{c_q^s:c_q^e}$, respectively. In the input W, the spans of those ACs are constructed as: $[w_{c_p^s:c_p^e}]$ and $[w_{c_q^s:c_q^e}]$. And, in the output, the augmented labels are created as: $[w_{c_q^s:c_q^e} \mid c_q \mid w_{c_p^s:c_p^e}]$ to make connection between the pairs of ACs, and for the head AC C_p, the augmented label is written as: $[w_{c_p^s:c_p^e} \mid c_p]$. The remaining tokens in W are rewritten as they are for both input and output. Finally, all the related key-phrase from all the ARs present in W are appended at the end of the label-augmented paragraph as:

`Strongly Related Key Phrases:` (e_{1_h}, e_{1_t}), (e_{2_h}, e_{2_t}), $\ldots$, (e_{k_h}, e_{k_t}).

Corpus	Model	Macro-F1		AVG
		ACC	ARI	
CDCP	Deep-Res-LG	65.3	29.3	47.3
	St-SVM (strict)	73.2	26.7	50.0
	TSP-PLBA	78.9	34.0	56.4
	BERT-Trans	82.5	37.3	59.9
	SB-Parser	82.3	40.1	61.2
	PITA	83.6	44.9	64.3
	Base **(Ours)**	77.9	35.3	56.6
	XL **(Ours)**	**84.2**	53.4	68.8
	XXL **(Ours)**	**84.2** (+0.6)	**53.9** (+9.0)	**69.0** (+4.7)
AAE-FG	SB-Parser*	56.6	67.8	62.2
	Base **(Ours)**	71.4	66.1	68.8
	XL **(Ours)**	**74.0** (+17.4)	**69.4**	**71.7** (+9.5)
	XXL **(Ours)**	73.3	**69.4** (+1.6)	71.3
AAE	Joint-ILP	82.6	58.5	70.6
	St-SVM (full)	77.6	60.1	68.9
	Joint-Ptr-Net	84.9	60.8	72.9
	Span-LSTM	85.7	67.8	76.8
	SB-Parser	86.8	69.3	78.1
	BERT-Trans	88.4	70.6	79.5
	MRC-GEN	89.2	70.9	80.1
	PITA	88.3	**73.5**	80.9
	Base **(Ours)**	87.4	69.3	78.4
	XL **(Ours)**	89.4	72.7	81.1
	XXL **(Ours)**	**89.5** (+0.3)	**73.5**	**81.5** (+0.6)

Table 1: Comparison of experimental results against the baselines. Best scores are marked in **bold**. Recent SoTA results are underlined. * indicates the baseline results produced by running the corresponding open-source code with original hyperparameters. All values are rounded to one decimal place. *Base*, *XL*, and *XXL* refer to different variants of the *Flan-T5* model.

4 Experimental Setup

4.1 Datasets

We evaluate our proposed method on three structurally distinct standard AM benchmarks: (i) **AAE**: Argument Annotated Essay (Stab and Gurevych, 2017), (ii) **AAE-FG**: Fine-Grained Argument Annotated Essay (Schaefer et al., 2023), and (iii) **CDCP**: Consumer Debt Collection Practices (Niculae et al., 2017). Both AAE and AAE-FG are tree-structured, while CDCP is non-tree-structured. Details of datasets are provided in Appendix B.

4.2 Implementation Details

We utilize the Flan-T5 model family (Chung et al., 2024) in three variants: *Base* (220M), *XL* (3B), and *XXL* (11B) for all our experiments. A batch size of 32 is used for AAE and AAE-FG, and that of 16 is used for CDCP. In each case a maximum input/output sequence length of 1024 is kept. We apply a learning rate of 0.0005 with the AdamW optimizer. All experiments are conducted on a single A100 GPU over 10000 steps, with checkpoints taken every 400 steps. The results are averaged over three independent runs. For the *XL* and *XXL* variants, we incorporate QLoRA adapters (Dettmers et al., 2024) for parameter-efficient fine-tuning. Further details on the QLoRA hyperparameters are given in Appendix A.

4.3 Evaluation

Following the prior studies (Bao et al., 2021; Liu et al., 2023), we evaluate the performance of the proposed method using the *Macro-Averaged F1* score for both ACC and ARI tasks. We consider only exact matches of AC spans as correct, disregarding any partial matches to maintain strict alignment with previous benchmarks.

4.4 Baselines

We consider the following SoTA joint models as baselines: **Joint-ILP** (Stab and Gurevych, 2017), **St-SVM** (Niculae et al., 2017), **Joint-Ptr-Net** (Potash et al., 2017), **Deep-Res-LG** (Galassi et al., 2018), **Span-LSTM** (Kuribayashi et al., 2019), **TSP-PLBA** (Morio et al., 2020), **BERT-Trans** (Bao et al., 2021), **SB-Parser** (Morio et al., 2022), **MRC-GEN** (Liu et al., 2023), and **PITA** (Sun et al., 2024). Details of these baselines are described in Appendix C.

5 Results and Discussion

5.1 Comparison with Baseline Models

Table 1 compares the proposed method with the SoTA baselines. Our approach consistently outperforms all existing baselines by a significant margin in both ACC and ARI tasks, achieving new SoTA results across all datasets. While most baselines perform well on the tree-structured AAE dataset, they struggle with the non-tree-structured CDCP dataset, particularly for the ARI task. The mix of transitive and non-transitive relations in CDCP pose challenges that our method handles relatively better. In this dataset, the proposed method marginally outperforms the latest baseline by 0.6% F1 score for the ACC task with an impressive relative improvement of F1 score of 9% for the challenging ARI task. Such improvements indicate the strength of our method in managing complex relationships. In the case of the tree-structured AAE dataset, it gave similar F1 scores or marginal improvement over the compared baselines for both the ACC and ARI tasks. This shows that the proposed approach works equally well for both tree-structured and non-tree-structured data. For the AAE-FG dataset, we benchmark our results against the *SB-Parser*

Error Type	Related AC Spans	Extracted Key Phrases	Correct Key Phrases
Generated New Words	AC_1: The better a person feels, the better his brain works. AC_2: Putting physical activities in early steps of human development would finally lead to mentally healthy society.	*(Better feelings, Mentally healthy society)*	*(Better his brain works, Mentally healthy society)*
Smaller or Larger Phrases	AC_1: The more cars and motorbikes are on roads, the more seriously the ozone layer is damaged. AC_2: This is sure to lead to more carbon emitted into the atmosphere, which can cause skin cancer.	*(Cars, Carbon emitted)*	*(Cars and motorbikes, Carbon emitted)*
Unrelated Phrases	AC_1: Roommate turns down the music or television volume at night time. AC_2: Consideration is always important in relationship.	*(Nighttime, Relationship)*	*(Roommate, Relationship)*

Table 2: Examples of different erroneous extraction of related key phrases using *Meta-LLaMa-3.1-instruct*.

Corpus	Variant	Macro-F1	
		ACC	ARI
CDCP	With KP	77.88	**35.28**
	Without KP	**80.58** (+2.7)	34.27 (-1.01)
AAE-FG	With KP	**71.39**	**66.13**
	Without KP	69.83 (-1.56)	64.07 (-2.06)
AAE	With KP	**87.44**	**69.26**
	Without KP	87.20 (-0.24)	68.02 (-1.24)

Table 3: Experimental results of *"with"* or *"without"* the related key phrases (KP) in the target ANL with *Flan-T5-Base*. Best scores are marked in **bold**.

baseline. Considering the original data content of AAE is classified into nine AC classes in AAE-FG as compared to three AC classes in the original one, this baseline struggles to classify those fine-grained AC classes correctly, resulting in poor performance in the ACC task. In contrast, our method achieves a significant 17.38% F1 score gain in ACC and a 1.65% F1 score improvement in ARI as compared to the baseline. Such strong performance of our ***ANL-based method*** across diverse datasets, whether tree-structured or not, highlights its generalizability to manage complex argumentative structures, all while achieving SoTA results.

5.2 Impact of Related Key Phrases

To assess the contribution of the related key phrases, we perform the proposed task *without using key phrase information* in the output ANL. For this, we use the *Flan-T5-Base* model. As shown in Table 3, omitting key phrases leads to a decline in performance in most of the cases across all datasets. This highlights the ***positive impact of key phrase information*** within the ANL in improving both ACC and ARI tasks. By generating this information, the model gains an internal understanding of the phrase-level AR connections, enabling it to better distinguish between related ACs. As a result, the likelihood of incorrectly associating non-related ACs is reduced, which significantly boosts ARI performance. Noticeably, the drop in ARI performance is more substantial than in ACC for all datasets. However, there is an exception: the CDCP dataset actually performs better *without key phrases* in the ACC task. This exception could be due to 207 out of 731 paragraphs lacking ARs, meaning a significant portion of the target ANL lacks key phrase information. This increases the difficulty for the model to generalize across both types of ANL sequences in the same training set: *those that contain key phrases* and *those that do not*. This complex distribution negatively impacts the ACC task performance when the *key phrase* information is on.

5.3 Human Evaluation of the Extracted Key Phrases

Since standard AM benchmarks lack explicit annotations for related key phrases, we conducted a manual evaluation. Two annotators (authors of this paper) assessed 10% of randomly selected related ACs and their key phrases from the AAE and CDCP training sets, classifying them as *Correct* or *Wrong*. Out of 506 key phrases analyzed, 451 and 442 were correct, yielding an average accuracy of **88.24%**. The main error types are summarized in Table 2.

Generated New Words: The model generates a new phrase instead of extracting it directly from the span, resulting in a mismatch with the original span words. This error is the most frequent one.

Smaller or Larger Phrases: The extracted spans are either too short or too long. Very few instances of this type of error were found.

Unrelated Phrases: The extracted phrases are not semantically related to the corresponding ACs.

We relied entirely on the capabilities of the LLM for related key phrase extraction tasks and did not manually filter the erroneous extractions, leaving them unchanged as they were utilized during training for the proposed generation task. Despite using these silver-standard key phrases, we achieved SoTA results, highlighting the strength of the proposed method even when the utilized key phrases are partially noisy.

KP Status	Corpus	Variants	Macro-F1	
			ACC	ARI
with KP	CDCP	Joint	77.88	**35.28**
		ACC Only	**79.71** (+1.83)	–
		ARI Only	–	33.73 (-1.55)
	AAE-FG	Joint	**71.39**	**66.13**
		ACC Only	70.29 (-1.1)	–
		ARI Only	–	65.06 (-1.07)
	AAE	Joint	**87.44**	**69.26**
		ACC Only	86.75 (-0.69)	–
		ARI Only	–	65.06 (-4.2)
w/o KP	CDCP	Joint	**80.58**	**34.27**
		ACC Only	79.83 (-0.75)	–
		ARI Only	–	33.22 (-1.05)
	AAE-FG	Joint	69.83	64.07
		ACC Only	69.83	–
		ARI Only	–	**66.12** (+2.05)
	AAE	Joint	**87.20**	**68.02**
		ACC Only	85.55 (-1.65)	–
		ARI Only	–	66.12 (-1.9)

Table 4: Joint Vs. Standalone formulation of AM tasks including *"with"* and *"without"* Key Phrases (KP) with *Flan-T5-Base*. Best scores are in **bold**.

Corpus	Variant	Macro-F1	
		ACC	ARI
CDCP	No Noise	**77.88**	35.28
	Noise-Inf	75.11 (-2.77)	28.29 (-6.99)
	Noise-FT	77.65 (-0.23)	**36.81** (+1.53)
AAE-FG	No Noise	71.39	**66.13**
	Noise-Inf	70.81 (-0.58)	60.89 (-5.24)
	Noise-FT	**71.48** (+0.09)	64.02 (-2.11)
AAE	No Noise	87.44	**69.26**
	Noise-Inf	86.14 (-1.3)	65.09 (-4.17)
	Noise-FT	**87.50** (+0.06)	68.02 (-1.24)

Table 5: Performance comparison under different noise setups using *Flan-T5-Base*. *No Noise* refers to training and inference without noise, *Noise-Inf* adds noise only during inference, and *Noise-FT* includes noise in both training and inference.

5.4 Joint vs. Standalone Formulation

To evaluate the importance of solving ACC and ARI tasks together rather than handling them independently, we propose two standalone task formulations by modifying the original ANL: *(i) ACC Only*, which focuses solely on the ACC task; *(ii) ARI Only*, which handles only the ARI task. In the *ACC Only* formulation, we remove the related AC span $w_{c_p^s:c_p^e}$ from the original augmented label $[w_{c_q^s:c_q^e} \mid c_q \mid w_{c_p^s:c_p^e}]$, creating a new augmented label $[w_{c_q^s:c_q^e} \mid c_q]$ in the target ANL sequence. In the *ARI Only* formulation, the AC type c_q is removed to make a new augmented label as $[w_{c_q^s:c_q^e} \mid w_{c_p^s:c_p^e}]$, which only contains the two related AC spans without their class labels. The input ANL format remains unchanged for both formulations. We formulate the target ANL in both *with* and *without* key phrase information settings.

Table 4 shows the joint setting outperforms the standalone formulations in nearly all cases. This demonstrates the advantage of joint modeling over treating the AM tasks independently. The most significant drop of 4.2% F1 scores was observed for the ARI task on AAE while using key phrases. Performance drop is also observed for the ACC task, with a maximum drop of 1.65% F1 scores on AAE. The joint formulation benefits from the mutual feature sharing across different tasks formulated in the same target ANL, uplifting each others' performances. In contrast, the standalone formulations miss out on these advantages, leading to lower performance. However, we observe two exceptions, one each in CDCP and AAE-FG. For the CDCP, the standalone model using key phrases performs better on the ACC task. This is likely due to the fact that 207 out of 731 paragraphs lack ARs in the joint formulation, complicating generalization across ANL sequences *with* and *without* ARs. In contrast, the exclusion of ARs in the standalone formulation enhances the performance in this dataset. In the AAE-FG dataset, the standalone ARI task yields better results than the joint formulation with muted key phrase information. The higher number of AC types in this dataset may lead to increased number of misclassified ACs, resulting in *error propagation* with inaccurate relation identification in the joint formulation. However, by removing AC-type information in the standalone ARI task, the model improves performance by reducing the chance of error propagation.

5.5 Noise Adaptability Study

This analysis aims to evaluate the robustness of our proposed method to noise by introducing noisy sentences into the input ANL. These noisy sentences are unrelated to the paragraph's content, often presenting an opposing viewpoint or being distinctly off-topic. We use few-shot prompting containing manually created noise with *Meta-LLaMa-3.1-instruct* to generate noisy sentences (See Appendix D). On average, the AAE and CDCP datasets contain 72.35 and 111.2 tokens per paragraph, respectively. Inserting a single noisy sentence in a paragraph adds roughly 18.57% noisy tokens to AAE and 17.2% to CDCP. For this significant amount of noise injection, we define two experimental setups: **(I)** *Train the model with noise and evaluate with noisy inputs.* **(II)** *Train the model without noise and evaluate with noisy inputs.* Table 5 presents the

Corpus	Model	Macro-F1		AVG
		ACC	ARI	
CDCP	**Flan-T5 (Fine-Tuned)**	**80.58**	**34.27**	**38.51**
	LLaMa *(5-shot)*	39.96	8.50	16.41
	LLaMa *(10-shot)*	52.95	8.88	20.82
	LLaMa *(20-shot)*	52.40	7.64	20.29
AAE-FG	**Flan-T5 (Fine-Tuned)**	**69.83**	**64.07**	**44.88**
	LLaMa *(5-shot)*	25.73	38.75	21.76
	LLaMa *(10-shot)*	37.61	40.95	26.40
	LLaMa *(20-shot)*	39.52	43.95	28.08
AAE	**Flan-T5 (Fine-Tuned)**	**87.20**	**68.02**	**51.97**
	LLaMa *(5-shot)*	43.56	19.80	21.37
	LLaMa *(10-shot)*	42.53	27.36	23.51
	LLaMa *(20-shot)*	42.46	39.59	27.62

Table 6: Performance comparison of Few-shot *Meta-LLaMa-3.1-instruct* Vs. Fine-tuned *Flan-T5-Base*.

Dataset	Model	2-length (%)	3-length (%)	4-length (%)	AVG (%)
CDCP	Base	6.52	3.57	0.00	3.36
	XL	**26.09**	**25.00**	**33.33**	**28.81**
	XXL	10.87	7.14	0.00	6.67
AAE-FG	Base	38.44	27.06	0.00	21.83
	XL	**46.26**	35.29	6.25	29.93
	XXL	43.54	**35.88**	**18.75**	**32.06**
AAE	Base	39.46	30.00	**18.75**	29.40
	XL	**50.68**	**40.59**	6.25	**32.51**
	XXL	43.88	33.53	12.50	29.97

Table 7: Performance analysis of *n-length* chains in terms of Accuracy(%), where $n = \{2, 3, 4\}$.

results under these conditions. The performance on the ACC task remains notably stable across datasets in both setups, highlighting the resilience of our method in noisy environments. However, for the ARI task, there is a significant drop in performance when noise is introduced during inference without prior exposure in training. When the model is trained with noise, the drop is notably smaller.

5.6 Few-shot Decoder-based LLM vs Fine-tuned Flan-T5-Base

Table 6 presents a comparison between the performance of 5-shot, 10-shot, and 20-shot experiments using the SoTA decoder-based LLM *Meta-LLaMa-3.1-instruct* and the fine-tuned *Flan-T5-Base*, both evaluated without key-phrase information. Although the few-shot performance improves as the number of examples increases, it consistently falls short of the fine-tuning approach, which outperforms it by a substantial margin across all datasets. It is worth noting that we did not conduct an extensive search for the optimal prompt in the AM task, as determining the most effective prompt can be quite challenging. Instead, we use the same input-output combination employed in the text-to-text generation of the proposed model to construct the few-shot prompts.

5.7 Argumentative Structural Analysis

We perform an in-depth performance study of our proposed approach in three important argumentative structural aspects as follows:

(I) Analysing the Relation Chains: Consider an argumentative paragraph containing multiple ACs, denoted as C_1, C_2, C_3, C_4, and C_5. These ACs may form a sequential chain such as $C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_4 \rightarrow C_5$, with C_5 being the root AC. Each consecutive AC is related to the previous one, thereby forming a *4-length* chain, as four relations connect these five ACs. Several such *n-length* chains are present in a paragraph in both AAE and CDCP, where $n = \{2, 3, 4\}$. In such configurations, relation identification between two closely connected ACs becomes challenging. Because, given that all ACs in the chain are (in-)directly related through the root, there is a high likelihood of incorrectly identifying relations, such as predicting $C_3 \rightarrow C_5$ instead of the correct $C_4 \rightarrow C_5$. Table 7 reports the performance of different length chains across different datasets with all three model variants. Considering the inherent difficulties, the results are promising in both tree and non-tree-structured datasets. As the chain length increases, the larger models mostly show better capability in detecting these chains correctly. Notably, we consider the counts of smaller sub-chains that are part of the bigger chains in our analysis.

(II) Capturing Long-Range Relations: In an argumentative paragraph, some ACs are related despite being far apart. For instance, one AC may appear at the start, while the other is found near the end. We compare the long-range relation identification performance based on the number of intermediate ACs between the linked components. Figure 3 shows a heatmap comparing model performance across varying distances. For both versions of the AAE dataset, most model variants excel at capturing these distant connections. However, in CDCP, the *Base* model struggles, and the *XL* model outperforms the *XXL* variant. Short-range relations with few or no intermediate ACs are more prevalent across the datasets, making them easier for most models to identify. Interestingly, the strong performance on long-range connections, despite their presence in lower numbers across datasets, showcases the effectiveness of our approach.

(III) Performance of Transitive Relations: The inherent challenge of the ARI task for non-tree-structured arguments lies in the presence of

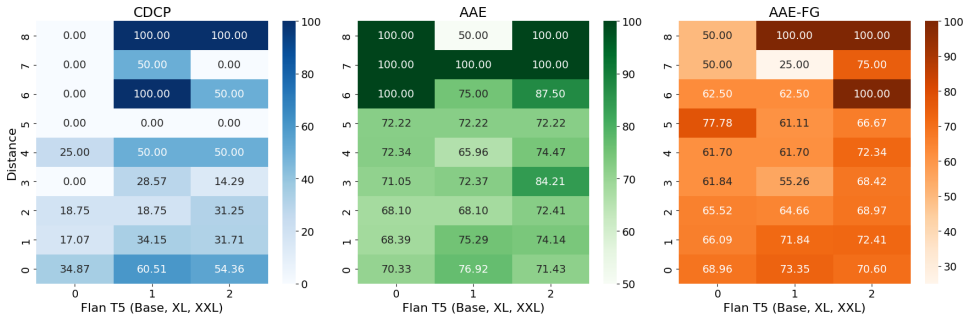

Figure 3: Performance comparison of the ARI task in capturing long-range relations, based on the distance between related ACs, measured by the number of components separating them. The X-axis represents different Flan-T5 variants *(Base, XL, XXL)*, while the Y-axis indicates the distance.

Ground Truth ANL	That is not to say, however, that advertisements have no downsides. Of course, [the advertising expenses lead to a higher product price and some of them express fake information, creating information asymmetry between consumers and companies | **Claim**]. Yet, [its merits still outweigh these downsides | **Premise** | the advertising expenses lead to a higher product price and some of them express fake information, creating information asymmetry between consumers and companies]. `Strongly Related Key Phrases: (Merits, Advertising expenses)`
Flan-T5-Base generated ANL	That is not to say, however, that advertisements have no downsides. Of course, [the advertising expenses lead to a higher product price and some of them express fake information, creating information asymmetry between consumers and companies | **Claim**]. Yet, [its merits still outweigh these downsides | **MajorClaim**].
Flan-T5-XXL generated ANL	That is not to say, however, that advertisements have no downsides. Of course, [the advertising expenses lead to a higher product price and some of them express fake information, creating information asymmetry between consumers and companies | **Claim**]. Yet, [its merits still outweigh these downsides | **Premise** | the advertising expenses lead to a higher product price and some of them express fake information, creating information asymmetry between consumers and companies]. `Strongly Related Key Phrases: (Merits, Advertisement expense)`

Table 8: Errors where missing key phrases in the generated sequence cause incorrect AC classification with *Flan-T5-base*, while the larger *Flan-T5-XXL* generates the correct ANL sequence. Incorrect generations are marked in red, and correct ones in green.

additional transitive relations. We evaluate the performance of identifying these transitive relations on the non-tree-structured CDCP corpus across different model variants. The *base* variant identified 9 out of 31 ground truth transitive relations, achieving an accuracy of 29.03%. The *XL* variant performed better, correctly identifying 14 out of 31 transitive relations with an accuracy of 45.16%. The *XXL* variant exhibited the best results, detecting 17 out of 31 transitive relations, yielding an accuracy of 54.84%. These findings highlight the strength of our method in handling complex non-tree argumentative structures.

5.8 Error Analysis

In all datasets, there are a few instances where the *Base* model either fails to capture or mistakenly captures relational dependencies. One such instance is shown in Table 8. This issue arises due to the erroneous generation or missing key phrases. In contrast, the larger *XXL* model effectively generates correct key phrases, enabling accurate identification of AC types and their spans, thus reducing errors. Also, a small number of cases exhibit unclosed brackets in the augmented labels in the generated ANL sequences. Since these errors are rare, we choose not to address them and retain the generated output as it is.

6 Conclusion

This paper introduces an ANL-based generative framework for jointly modeling ACC and ARI tasks in AM. By utilizing few-shot LLM prompting to extract related key phrases between AC pairs, we enrich the output ANL to improve the capability of capturing complex relations. With extensive experiments across multiple datasets, the proposed approach achieves SoTA results and handles both tree- and non-tree-structured data with ease, demonstrating strength and generalizability. Moreover, it handles noise effectively and is capable of capturing longer relation chains, long-range relational dependencies and transitive relations.

7 Limitations and Future Scope

Despite the strong performance of our ANL-based method for AM tasks, several limitations remain. First, the accuracy of the extracted key phrases is closely tied to the quality of the prompts and the

performance of the LLM. As the LLM's output is not fully predictable, it can occasionally generate irrelevant or incomplete key phrases. These are discarded, and we do not attempt further extraction from these instances, leaving them empty during the construction of the proposed target ANL. Second, the method's reliance on key phrases introduces variability in performance, particularly in datasets like CDCP, where the inclusion of key phrases sometimes leads to performance drops. A more dynamic strategy for key phrase integration, tailored to dataset characteristics, could improve results. Another limitation is the challenge of handling transitive relations in non-tree-structured arguments. While our method performs well, the accuracy for detecting transitive links remains moderate (up to 54.84%), indicating room for improvement in more complex, non-hierarchical structures. Finally, the joint modeling of ACC and ARI tasks can lead to error propagation, especially in datasets with a large number of AC types, such as AAE-FG, where mistakes in one task may negatively impact the other. Refining error mitigation strategies could help reduce these issues. Addressing these limitations in future work will enhance the robustness and adaptability of the proposed method further.

Acknowledgments

We thank the anonymous reviewers for their valuable feedback. This work was supported in part by the Ministry of Human Resource Development (MHRD), Govt. of India, for financial assistance. This work is also supported by the IITG Technology Innovation and Development Foundation (TI&DF) as a part of the National Mission on Interdisciplinary Cyber-Physical Systems with financial assistance from the Department of Science and Technology, India, through grant number DST/NMICPS/TIH12/IITG/2020.

References

Ben Athiwaratkun, Cicero Nogueira dos Santos, Jason Krone, and Bing Xiang. 2020. Augmented Natural Language for Generative Sequence Labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 375–385.

Jianzhu Bao, Chuang Fan, Jipeng Wu, Yixue Dang, Jiachen Du, and Ruifeng Xu. 2021. A neural transition-based model for argumentation mining. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6354–6364, Online. Association for Computational Linguistics.

Tuhin Chakrabarty, Aadit Trivedi, and Smaranda Muresan. 2021. Implicit premise generation with discourse-aware commonsense knowledge models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6247–6252, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Saadia Gabriel, Chandra Bhagavatula, Vered Shwartz, Ronan Le Bras, Maxwell Forbes, and Yejin Choi. 2021. Paragraph-level commonsense transformers with recurrent memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12857–12865.

Andrea Galassi, Marco Lippi, and Paolo Torroni. 2018. Argumentative link prediction using residual networks and multi-objective learning. In *Proceedings of the 5th Workshop on Argument Mining*, pages 1–10, Brussels, Belgium. Association for Computational Linguistics.

Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2020. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. In *AAAI Conference on Artificial Intelligence*.

Guochao Jiang, Ziqin Luo, Yuchen Shi, Dixuan Wang, Jiaqing Liang, and Deqing Yang. 2024. Toner: Type-oriented named entity recognition with generative language model. In *International Conference on Language Resources and Evaluation*.

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, St. Julian's, Malta. Association for Computational Linguistics.

Zixuan Ke, Winston Carlile, Nishant Gurrapadi, and Vincent Ng. 2018. Learning to give feedback: Modeling attributes affecting argument persuasiveness in student essays. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4130–4136. International Joint Conferences on Artificial Intelligence Organization.

Tatsuki Kuribayashi, Hiroki Ouchi, Naoya Inoue, Paul Reisert, Toshinori Miyoshi, Jun Suzuki, and Kentaro Inui. 2019. An empirical study of span representations in argumentation structure parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4691–4698, Florence, Italy. Association for Computational Linguistics.

Boyang Liu, Viktor Schlegel, Riza Batista-Navarro, and Sophia Ananiadou. 2023. Argument mining as a multi-hop generative machine reading comprehension task. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10846–10858, Singapore. Association for Computational Linguistics.

Tianyu Liu, Yuchen Jiang, Nicholas Monath, Ryan Cotterell, and Mrinmaya Sachan. 2022. Autoregressive structured prediction with language models. In *Conference on Empirical Methods in Natural Language Processing*, pages 993–1005.

Tobias Mayer, Elena Cabrio, and Serena Villata. 2020. Transformer-based argument mining for healthcare applications. In *European Conference on Artificial Intelligence*.

Gaku Morio, Hiroaki Ozaki, Terufumi Morishita, Yuta Koreeda, and Kohsuke Yanai. 2020. Towards better non-tree argument mining: Proposition-level biaffine parsing with task-specific parameterization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3259–3266, Online. Association for Computational Linguistics.

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.

Vlad Niculae, Joonsuk Park, and Claire Cardie. 2017. Argument Mining with Structured SVMs and RNNs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 985–995.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. *ArXiv*, abs/2101.05779.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2017. Here's my point: Joint pointer architecture for argument mining. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1364–1373, Copenhagen, Denmark. Association for Computational Linguistics.

Ameer Saadat-Yazdi, Jeff Z. Pan, and Nadin Kokciyan. 2023. Uncovering implicit inferences for improved relational argument mining. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2484–2495, Dubrovnik, Croatia. Association for Computational Linguistics.

Robin Schaefer, René Knaebel, and Manfred Stede. 2023. Towards fine-grained argumentation strategy analysis in persuasive essays. In *Proceedings of the 10th Workshop on Argument Mining*, Singapore.

Christian Stab and Iryna Gurevych. 2017. Parsing Argumentation Structures in Persuasive Essays. *Computational Linguistics*, 43(3):619–659.

Yang Sun, Bin Liang, Jianzhu Bao, Min Yang, and Ruifeng Xu. 2022. Probing structural knowledge from pre-trained language model for argumentation relation classification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3605–3615, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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, Bangkok, Thailand. Association for Computational Linguistics.

Dietrich Trautmann. 2020. Aspect-based argument mining. In *Proceedings of the 7th Workshop on Argument Mining*, pages 41–52, Online. Association for Computational Linguistics.

Somin Wadhwa, Silvio Amir, and Byron Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15566–15589, Toronto, Canada. Association for Computational Linguistics.

Vern R. Walker, Dina Foerster, Julia Monica Ponce, and Matthew Rosen. 2018. Evidence types, credibility factors, and patterns or soft rules for weighing conflicting evidence: Argument mining in the context of legal rules governing evidence assessment. In *Proceedings of the 5th Workshop on Argument Mining*, pages 68–78, Brussels, Belgium. Association for Computational Linguistics.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021. Towards Generative Aspect-Based Sentiment Analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 504–510.

A Hyperparameters

We use the following hyperparameters setting for fine-tuning with QLoRA:

Parameter	Value
r	16
lora alpha	32
lora dropout	0.05
bias	none
task type	SEQ_2_SEQ_LM
target modules	$q, v, k, o, wo, wi_0, wi_1$
load_in_4bit	True
bnb_4bit_quant_type	nf4
bnb_4bit_use_double_quant	True
bnb_4bit_compute_dtype	torch.bfloat16

Table 9: Hyperparameter setting of QLoRA

B Dataset Description

We conduct experiments on three standard AM datasets that are structurally distinct. A brief description of these datasets is given below.

Argument Annotated Essay (AAE) (Stab and Gurevych, 2017): This dataset comprises a tree-structured annotation scheme, where each AC consists of at most one outgoing AR. It contains 402 student essays annotated at the segment (span) level. Each essay is organized into several paragraphs. In total, there are 1,833 paragraphs containing three types of ACs: *Claim*, *MajorClaim*, and *Premise*, resulting in 6,089 ACs and 3,832 ARs.

Fine-Grained Argument Annotated Essay (AAE-FG) (Schaefer et al., 2023): It is a more detailed annotation of the AAE dataset, where the ACs are categorized in a fine-grained manner. The *Major Claim* and *Claim* categories have been subdivided into *Fact*, *Value*, and *Policy*. Similarly, the *Premise* category has been further refined into *Common Ground*, *Testimony*, *Hypothetical Instance*, *Statistics*, *Real Example*, and *Others*. Consequently, there are now nine AC types in total: *Fact*, *Value*, *Policy*, *Common Ground*, *Testimony*, *Hypothetical Instance*, *Statistics*, *Real Example*, and *Others*.

Consumer Debt Collection Practices (CDCP) (Niculae et al., 2017): This dataset is annotated with a non-tree argumentation scheme, where an AC might contain more than one outgoing AR. It contains 731 user comments from the Consumer Financial Protection Bureau (CFPB) website. It includes five AC types: *Fact*, *Testimony*, *Reference*, *Policy*, and *Value* with two AR types: *Reason* and *Evidence*. A total of 4931 ACs are present in this dataset, while the number of ARs is 1220.

C Details of Baselines

Joint-ILP (Stab and Gurevych, 2017): An argumentation structure parser that performs joint optimization of ACs and ARs using Integer Linear Programming (ILP).
St-SVM (Niculae et al., 2017): A structured SVM that models AM as an inference problem in both full and strict factor graph.
Joint-Ptr-Net (Potash et al., 2017): This joint model leverages a Pointer Network architecture to simultaneously address ACC and ARI tasks.
Deep-Res-LG (Galassi et al., 2018): It utilizes a residual network with link-guided training.
Span-LSTM (Kuribayashi et al., 2019): An LSTM-based span representation with argumentative markers for improved AC and AR processing.
TSP-PLBA (Morio et al., 2020): It employs task-specific parameterization for encoding ACs and a biaffine attention module for ARs.
BERT-Trans (Bao et al., 2021): A neural model that employs a transition-based approach.
SB-Parser (Morio et al., 2022): Latest dependency parsing approach using Longformer combined with a Span-Biaffine(SB) architecture for sub-task information sharing.
MRC-GEN (Liu et al., 2023): A multi-hop generative formulation converting the AM tasks into machine reading comprehension.
PITA (Sun et al., 2024): A generative multitask formulation with prompt-tuning for efficient task interactions.

D Few-shot Prompt Template for Noise Generation

We present the following instance of a few-shot prompt to generate noisy sentences. In this prompt, the noisy sentences are carefully crafted by hand.
Input Prompt:

```
Given an argumentative paragraph, generate a noisy sentence in 5-20 words only, which contradicts this paragraph. Here are the following examples for your reference.

Paragraph: Technology has brought significant advantages to education. With the help of modern gadgets and the internet, students now have access to unlimited resources that can enhance their learning experience. Online courses and educational apps provide flexibility, allowing students to study
```

at their own pace and revisit difficult topics. Technology also encourages collaboration through platforms like Google Classroom, where students can work together on assignments remotely. Furthermore, interactive learning tools such as simulations and educational games make complex subjects more understandable and engaging. As technology continues to advance, its role in education will only grow, making learning more accessible and effective.

Noisy Sentence: Technology has no positive impact on education or learning.

Paragraph: Exercise is essential for maintaining good health. Regular physical activity helps to reduce the risk of many chronic diseases, including heart disease, diabetes, and obesity. It also improves mental health by reducing symptoms of depression and anxiety while boosting overall mood. Moreover, exercise promotes better sleep and increases energy levels, making daily tasks easier to manage. Whether it's running, swimming, or practicing yoga, staying active is a key factor in living a long and healthy life. Everyone should incorporate at least 30 minutes of exercise into their daily routine for optimal well-being.

Noisy Sentence: Exercise is unnecessary and offers no health benefits at all.

Paragraph: Sustainable energy is the key to a better future. With climate change threatening the planet, we must transition from fossil fuels to renewable energy sources like solar, wind, and hydropower. These alternatives produce far fewer greenhouse gases, reducing our carbon footprint and helping to combat global warming. Additionally, sustainable energy is cost-effective in the long run, as renewable resources are abundant and will not run out. By investing in renewable energy infrastructure today, we can create a cleaner, healthier environment for future generations while also fostering economic growth in green industries.

Noisy Sentence: Sustainable energy has no effect on the environment and isn't worth pursuing.

Paragraph: Reading books is one of the best ways to develop the mind and enhance knowledge. Books offer a wealth of information, provide deep insights, and stimulate critical thinking. Unlike short online articles or videos, books allow for a comprehensive exploration of subjects, fostering a greater understanding of various topics. Furthermore, reading improves focus, concentration, and vocabulary, all of which contribute to better communication skills. Whether fiction or non-fiction, books open up new worlds and perspectives, making them indispensable tools for intellectual growth and personal development.

Noisy Sentence: Reading books wastes time and doesn't improve knowledge or skills.

Paragraph: Healthy eating is crucial for maintaining a balanced lifestyle and preventing disease. A diet rich in fruits, vegetables, whole grains, and lean proteins provides the body with essential nutrients that support overall well-being. Proper nutrition strengthens the immune system, boosts energy levels, and reduces the risk of chronic conditions such as heart disease, diabetes, and obesity. Moreover, healthy eating improves mental clarity and mood, promoting both physical and psychological health. By making mindful food choices, we can ensure a healthier, longer life.

Noisy Sentence: Healthy eating has no effect on health and is unnecessary for well-being.

Real Example

Paragraph: In addition, being free to stay away from classes improves flexibility and therefore quality of student life. Sometimes the wild party on Thursday night is too good to end already at midnight only because of a lecture on Friday in the morning. With a liberal policy students are able to postpone the learning to the afternoon which gives a feeling of freedom and improves time efficiency. Research has shown that the more satisfied the students are with those

life aspects, the better they perform in academic areas.

Noisy Sentence:

Note: Give the sentence only without any prefix. No code please.

Output:

Being required to attend classes actually leads to better grades and higher satisfaction rates among students.