UniASA: A Unified Generative Framework for Argument Structure Analysis

Jianzhu Bao[1], Mohan Jing[2], Kuicai Dong[3], Aixin Sun[3],
Yang Sun[1], and Ruifeng Xu[1,4,5*]

[1]Harbin Institute of Technology (Shenzhen), China
jianzhubao@gmail.com, xuruifeng@hit.edu.cn

[2]Tsinghua University, China

[3]Nanyang Technological University, Singapore

[4]Peng Cheng Laboratory, China

[5]Guangdong Provincial Key Laboratory of
Novel Security Intelligence Technologies, China

Argumentation is a fundamental human activity that involves reasoning and persuasion, which also serves as the basis for the development of AI systems capable of complex reasoning. In NLP, to better understand human argumentation, argument structure analysis aims to identify argument components, such as claims and premises, and their relations from free text. It encompasses a variety of divergent tasks, such as end-to-end argument mining, argument pair extraction, and argument quadruplet extraction. Existing methods are usually tailored to only one specific argument structure analysis task, overlooking the inherent connections among different tasks. We observe that the fundamental goal of these tasks is similar: identifying argument components and their interrelations. Motivated by this, we present a unified generative framework for argument structure analysis (UniASA). It can uniformly address multiple argument structure analysis tasks in a sequence-to-sequence manner. Further, we enhance UniASA with a multi-view learning strategy based on subtask decomposition. We conduct experiments on seven datasets across three tasks. The results indicate that UniASA can address these tasks uniformly and achieve performance that is either superior to or comparable with the previous state-of-the-art methods. Also, we show that UniASA can be effectively integrated with large language models, such as Llama, through fine-tuning or in-context learning.

[*] Corresponding author

Action Editor: Preslav Nakov. Submission received: 8 February 2024; revised version received: 6 December 2024; accepted for publication: 10 January 2025.

https://doi.org/10.1162/coli_a_00553

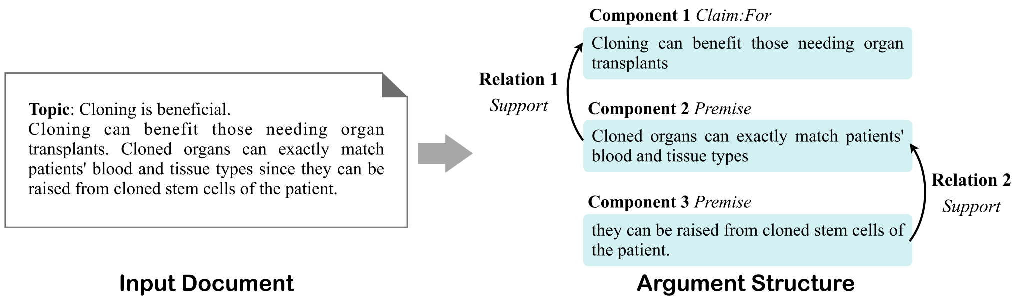
1. Introduction

Argumentation, or debate, is a fundamental human activity for persuading others or resolving conflicting views. It involves verbal, social, and rational interaction of ideas, with the goal of proving or refuting a particular viewpoint through a series of propositions (Van Eemeren, Grootendorst, and Grootendorst 2004). This process is vital in human communication as it facilitates reasoned discussions, alleviates conflicts, and fosters mutual understanding. The ability to understand and generate arguments underlies the development of artificial intelligence (AI) systems, such as large language models, enabling them to perform complex reasoning, support decision-making, and enhance human–computer interaction.

Computational argumentation, which involves analyzing and modeling human argumentation through computational methods, is an essential and challenging area within NLP (Lawrence and Reed 2019; Lauscher et al. 2022). It is beneficial to many domains, such as education (Song et al. 2020; Weber et al. 2023), law (Palau and Moens 2009; Elaraby, Zhong, and Litman 2023), and debate systems (Hua, Hu, and Wang 2019; Mestre et al. 2021), to name a few. Argument structure analysis is a key aspect of computational argumentation, involving the identification of **argument components** (e.g., claim, premise) and **argumentative relations** (e.g., support, attack) from free text. Through argument structure analysis, the argument structure embedded within the input argumentative document can be parsed out. Understanding argument structure is vital for advancing AI, particularly in the development of large language models. To achieve strong reasoning capabilities, these models must be able to comprehend and utilize argument structures effectively.

In order to analyze argument structure, there are several related tasks in existing studies, such as **end-to-end argument mining (E2E-AM)** (Kuribayashi et al. 2019; Bao et al. 2021a), **argument pair extraction (APE)** (Cheng et al. 2020; Sun et al. 2023), and **argument quadruplet extraction (AQE)** (Guo et al. 2023). The E2E-AM task generally focuses on (i) recognizing the spans of argument components at word-level, (ii) classifying the types of these components, and (iii) classifying the argumentative relations between them. Similarly, the AQE task identifies claims and their corresponding evidence at the sentence level (Guo et al. 2023). Different from the E2E-AM and AQE tasks, the APE task aims to identify and extract pairs of interrelated arguments from two documents (Cheng et al. 2021; Sun et al. 2023). For the sake of simplicity, the "claim" and "evidence" in the AQE task, along with the "argument" in the APE task, are all termed as "argument component" or simply "component" in this article. Figures 1, 2, and 3 illustrate examples of the E2E-AM, APE, and AQE tasks, respectively.

We summarize the characteristics and definitions for different argument structure analysis tasks and datasets in Table 1. According to their definitions, all of these tasks aim to analyze the structure of argumentation by identifying argument components and their interrelations. However, in practice, these argument structure analysis tasks involve identifying argument components at different levels of granularity, and classifying component and relation types according to different schemas. As a result, most argument structure analysis systems are tailored to specific tasks, resulting in isolated models with dedicated architectures. For example, many previous studies particularly focus on the E2E-AM task (Morio et al. 2022; Bao et al. 2021a) or its related subtasks (Stab and Gurevych 2014; Trautmann et al. 2020). Some other studies are committed to proposing models specifically designed for the APE task (Cheng et al. 2021; Sun et al. 2023). Recently, Guo et al. (2023) develop a generative model for the AQE task. The divergence of these task-specific solutions poses significant obstacles to architectural



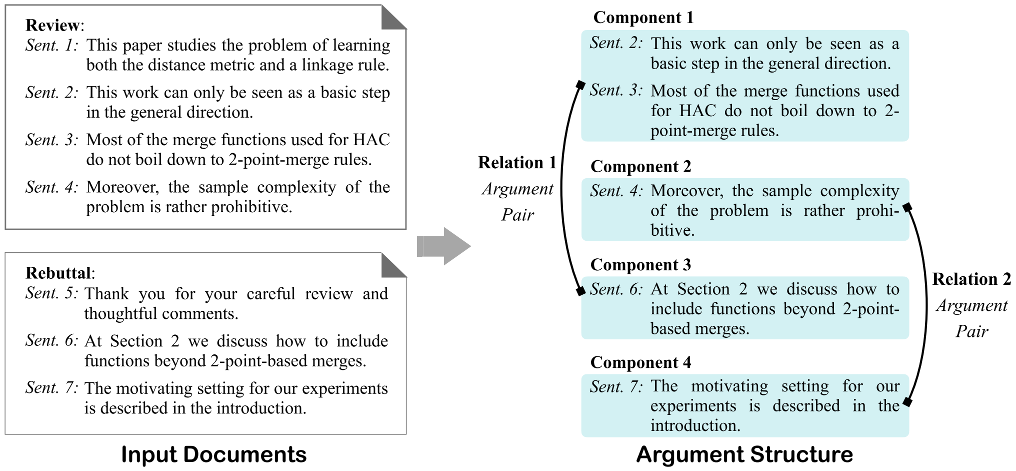
Topic: Cloning is beneficial.
Cloning can benefit those needing organ transplants. Cloned organs can exactly match patients' blood and tissue types since they can be raised from cloned stem cells of the patient.

Component 1 *Claim:For*
Cloning can benefit those needing organ transplants

Component 2 *Premise*
Cloned organs can exactly match patients' blood and tissue types

Component 3 *Premise*
they can be raised from cloned stem cells of the patient.

Figure 1
A simplified example of the end-to-end argument mining (E2E-AM) task. "*Claim:For*" and "*Premise*" represent the types of argument components, while "*Support*" represents one type of argumentative relation.



Review:
Sent. 1: This paper studies the problem of learning both the distance metric and a linkage rule.
Sent. 2: This work can only be seen as a basic step in the general direction.
Sent. 3: Most of the merge functions used for HAC do not boil down to 2-point-merge rules.
Sent. 4: Moreover, the sample complexity of the problem is rather prohibitive.

Rebuttal:
Sent. 5: Thank you for your careful review and thoughtful comments.
Sent. 6: At Section 2 we discuss how to include functions beyond 2-point-based merges.
Sent. 7: The motivating setting for our experiments is described in the introduction.

Component 1
Sent. 2: This work can only be seen as a basic step in the general direction.
Sent. 3: Most of the merge functions used for HAC do not boil down to 2-point-merge rules.

Component 2
Sent. 4: Moreover, the sample complexity of the problem is rather prohibitive.

Component 3
Sent. 6: At Section 2 we discuss how to include functions beyond 2-point-based merges.

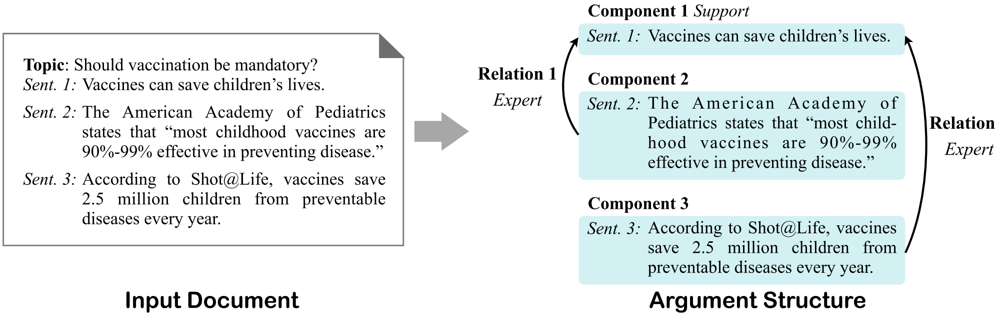
Component 4
Sent. 7: The motivating setting for our experiments is described in the introduction.

Figure 2
A simplified example of the argument pair extraction (APE) task. "*Sent. i*" represents the i-th sentence in the input text. "*Argument Pair*" indicates that the two argument components connected by this argumentative relation can form an argument pair.

development, knowledge sharing, and cross-domain adaptation for argument structure analysis tasks.

Recently, the paradigm of using a unified generative framework across diverse tasks has become a new trend in various research areas (Yan et al. 2021a; Lu et al. 2022). Such a paradigm offers several advantages. Specifically, (1) the knowledge and datasets from different tasks can be shared through this unified framework, thereby alleviating the issue of insufficient data for certain tasks; (2) a unified framework can eliminate the laborious need for designing isolated model architectures for each task. Consequently, we are inspired to design a straightforward unified framework for all argument structure analysis tasks.

To this end, we present a **Uni**fied generative framework for **A**rgument **S**tructure **A**nalysis (UniASA). First, we design a unified task template to formulate divergent argument structure analysis tasks into a uniform natural language representation. Specifically, the template includes: (i) a source sequence template that incorporates



Topic: Should vaccination be mandatory?
Sent. 1: Vaccines can save children's lives.
Sent. 2: The American Academy of Pediatrics states that "most childhood vaccines are 90%-99% effective in preventing disease."
Sent. 3: According to Shot@Life, vaccines save 2.5 million children from preventable diseases every year.

Component 1 *Support*
Sent. 1: Vaccines can save children's lives.

Component 2
Sent. 2: The American Academy of Pediatrics states that "most childhood vaccines are 90%-99% effective in preventing disease."

Component 3
Sent. 3: According to Shot@Life, vaccines save 2.5 million children from preventable diseases every year.

Figure 3
A simplified example of the argument quadruplet extraction (AQE) task. "*Support*" indicates that the argument component holds a supportive stance towards the input topic. "*Expert*" indicates that the argument component serves as an "Expert" type of evidence for the target component it supports.

the task instruction, input topic, and input text, and (ii) a target sequence template that flattens the argument structure into a text sequence. Subsequently, the formulated source and target sequences can be universally modeled by T5, a pre-trained sequence-to-sequence model (Raffel et al. 2020). Moreover, we design a multi-view learning strategy to enhance this unified framework. This strategy decomposes each argument structure analysis task into two subtasks: *argument component identification* and *argumentative relation identification*. The original argument structure analysis task identifies components and relations together, while the two subtasks identify them separately. Our multi-view learning strategy considers these two subtasks, along with the original argument structure analysis task, as three distinct learning perspectives. Through view-specific templates, UniASA can effectively learn and aggregate patterns from different views, leading to improved performance.

To evaluate the validity of UniASA, we carry out extensive experiments on seven datasets of three argument structure analysis tasks (Table 1). UniASA achieves results that are superior to or comparable with the previous state-of-the-art (SOTA) methods. Experiments in low-resource scenarios further reveal the validity and robustness of UniASA. We also demonstrate another advantage of UniASA in consolidating various argument structure analysis tasks: the potential for enhancement via intermediate fine-tuning. Furthermore, we show that UniASA can easily be integrated with large language models like Llama (Touvron et al. 2023) via fine-tuning or in-context learning. The results indicate that UniASA, when fine-tuned with a larger generative model, can achieve much greater performance improvements.

We summarize our contributions made in this article as follows:

- We introduce UniASA, a unified generative framework for argument structure analysis that can universally model different tasks.[1]
- We devise a multi-view learning strategy based on subtask decomposition to improve the capabilities of UniASA.

[1] Code: `https://github.com/HITSZ-HLT/UniASA`.

Table 1
A comparison of the characteristics of different argument structure analysis tasks in their definitions and datasets. "Doc." is short for "document". "Component" and "Relation" stand for "argument component" and "argumentative relation", respectively. For the AQE task, in order to align with other argument structure analysis tasks, we consider an argument component's stance ("Support", "Attack") towards the topic as the component category, while the support category ("Expert", "Research", "Case", "Explanation", "Others") of a component towards another component is regarded as the relation category.

Task	Dataset	Input	Output		
			Component Granularity	Component Category	Relation Category
E2E-AM	AAEC	1 Doc.	Word-level Span	{MajorClaim, Claim:For, Claim:Against, Premise}	{Support, Attack}
	CDCP			{Value, Fact, Policy, Testimony, Reference}	{Reason, Evidence}
	AbstRCT			{MajorClaim, Evidence, Claim}	{Support, Attack, Partial-Attack}
	MTC			{Opponent, Proponent}	{Example, Support, Undercut, Rebut}
	AASD			{Proposal, Means, Observation, Result, Assertion, Description}	{Support, Attack, Detail, Sequence, Additional}
APE	RR	2 Doc.	Sentence-level Span	{Argument}	{Argument Pair}
AQE	QAM	1 Doc.	Sentence	{Support, Against}	{Expert, Research, Case, Explanation, Others}

- We conduct experiments on seven datasets related to argument structure analysis, each of which has divergent annotation schemas. Experiments and further analysis indicate the effectiveness of UniASA.

2. Background

2.1 Argument Structure Theory

Argument structure is a crucial concept in the studies of logic and argumentation theory (Walton 1996; Freeman 2011). Through argument structure, one can easily see how the various components of an argument come together to back up a claim. In the field of informal logic, scholars have long been studying how to design argument structures to better understand and interpret human argumentation (Beardsley 1950; Thomas 1973). A classic approach to depicting argument structures is the Toulmin Model (Toulmin 1958). Freeman (2011) further refines the Toulmin Model, developing another classic method for illustrating argument structures. The development of computational argumentation, especially in argument structure analysis, has been largely influenced by these foundational argument structure theories (Peldszus 2014; Stab and Gurevych 2017).

2.2 End-to-end Argument Mining

Building on argument structure theories, many researchers explore computational approaches for the analysis of argument structures, initially referred to as argument(ation) mining (Lawrence and Reed 2019). To this end, a variety of datasets (Peldszus and Stede 2015; Stab and Gurevych 2017; Park and Cardie 2018; Mayer, Cabrio, and Villata 2020) and methods (Eger, Daxenberger, and Gurevych 2017; Ye and Teufel 2021; Bao et al. 2022a) have been proposed.

Datasets for the E2E-AM Task. Stab and Gurevych (2017) design a tree-structured annotation scheme for student argumentative essays and construct the Argument-annotated Essays Corpus (AAEC). Drawing from Freeman's theory (Freeman 2011), Peldszus and Stede (2015) annotate argument structures on manually written short texts, presenting the Microtexts (MTC) dataset. Mayer, Cabrio, and Villata (2020) propose the AbstRCT dataset for E2E-AM in the clinical domain. Park and Cardie (2018), by integrating various argument structure theories, propose an annotation scheme for online user comments, leading to the creation of the Consumer Debt Collection Practices (CDCP) dataset. Accuosto, Neves, and Saggion (2021) present a tree-structured dataset for the E2E-AM task based on abstracts of scientific papers (AASD).

Methods for the E2E-AM Task. The E2E-AM task encompasses two key subtasks: argument component identification and argumentative relation identification (Persing and Ng 2016). Many previous efforts have been dedicated to exploring a specific subtask independently (Stab and Gurevych 2014; Cocarascu and Toni 2017). For argument component identification, Chernodub et al. (2019) propose a sequence tagging method. Wang et al. (2020) develop a multi-scale approach for identifying argument components from coarse to fine. For argumentative relation identification, Jo et al. (2021) investigate multiple logical and theory-informed mechanisms to show the connections between the argumentative relations and the knowledge such as fact, sentiment, causality, etc. Saadat-Yazdi, Pan, and Kökciyan (2023) determine argumentative relations by generating the commonsense reasoning chains. Sun et al. (2022) improve the classification of argumentative relations by leveraging probed knowledge from pre-trained language models. Recently, there has been a growing number of studies focusing on the joint modeling of component and relation identification (Persing and Ng 2016; Potash, Romanov, and Rumshisky 2017; Kuribayashi et al. 2019; Morio et al. 2020). To this end, Bao et al. (2021a) design a transition-based approach for the incremental parsing of argumentation graphs. Building on a previous dependency parsing-based method (Eger, Daxenberger, and Gurevych 2017), Ye and Teufel (2021) refine the strategy for constructing dependency trees, resulting in improved performance. Bao et al. (2022a) introduce a generative approach equipped with a constrained pointer mechanism. Morio et al. (2022) develop a cross-corpora approach based on multi-task learning, aiming to mitigate the data sparsity issue in the E2E-AM task. Liu et al. (2023) present a multi-turn question answering approach for the E2E-AM task, which incorporates global argument structure information.

2.3 Argument Pair Extraction

Proposed by Cheng et al. (2020), the APE task is designed to extract pairs of arguments discussing the same point from two interrelated documents.

Datasets for the APE Task. Cheng et al. (2020) construct a large-scale benchmark for the APE task, known as the Review-Rebuttal (RR) dataset. This dataset is sourced from the reviews and rebuttals of the ICLR conference.

Methods for the APE Task. To address the APE task, Cheng et al. (2021) use a multi-cross encoding framework based on table-filling. Bao et al. (2021b) propose a mutual guided framework to address this task. Bao et al. (2022b) approach this task with a multi-turn machine reading comprehension model. Zhu et al. (2023) devise a multi-scale relation-aware graph to capture inter-sentence relations. Sun et al. (2023) approach the APE task by extracting semantic knowledge from pre-trained models using probing techniques.

Furthermore, another task closely related to argument pair extraction is the interactive argument pair identification task (Yuan et al. 2021a). This task is a multiple-choice task aimed at selecting replies that have an interactive relationship with a given argument. To solve this task, researchers explore utilizing discrete representation (Ji et al. 2021) and knowledge graphs (Yuan et al. 2021b).

2.4 Argument Quadruplet Extraction

Similar to the objective of the E2E-AM task, the AQE task (Guo et al. 2023) is defined as extracting claims and their evidence from a document, aiming to analyze its argument structure.

Datasets for the AQE Task. As a new task, the QAM dataset (Guo et al. 2023) currently stands as the only dataset for the AQE task. It is gathered from English Wikipedia and encompasses a wide range of debate topics.

Methods for the AQE Task. Based on T5 (Raffel et al. 2020), Guo et al. (2023) propose a quad-tagging augmented generative framework to address the AQE task.

2.5 Unified Generative Framework

In recent years, with the development of pre-training techniques, designing a unified generative framework to address multiple tasks within a specific field has garnered increasing attention (Yan et al. 2021a,b; Li et al. 2021; Lu et al. 2022; Li et al. 2023). For example, some researchers are dedicated to using a unified generative method to tackle various named entity recognition subtasks (Yan et al. 2021b; Lu et al. 2023; Zhang et al. 2023), or various information extraction tasks (Lu et al. 2022). In the field of sentiment analysis, some studies aim to address all aspect-based sentiment analysis subtasks (Yan et al. 2021a; Gao et al. 2022), all multimodal sentiment analysis tasks (Hu et al. 2022), or even all sentiment analysis-related tasks (Li et al. 2023). Moreover, such unified generative frameworks also demonstrate promising results in other fields, such as dialogue understanding (Chen et al. 2022; He et al. 2022) and multimodal summarization (Zhang et al. 2022).

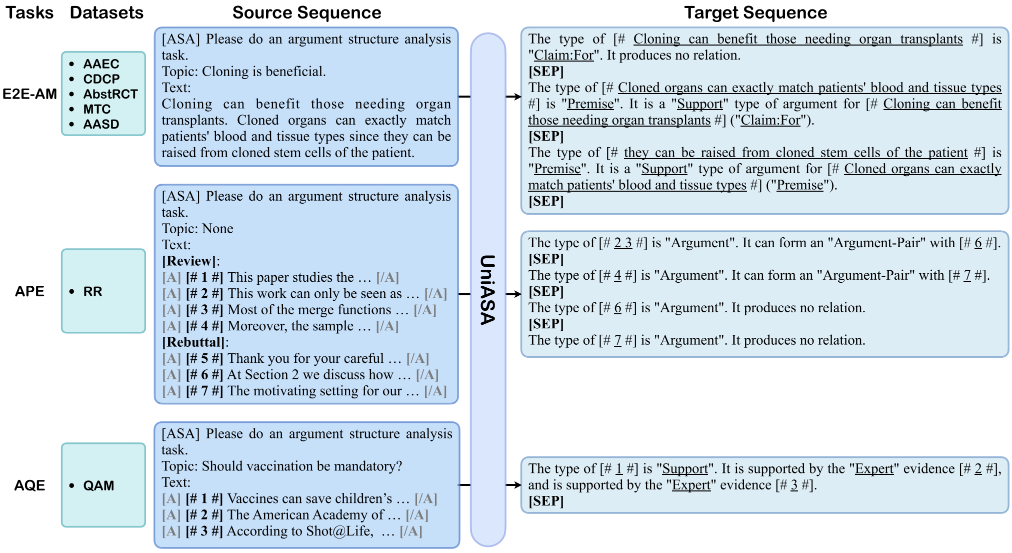
In this article, we present UniASA, a unified generative framework specifically tailored for various argument structure analysis tasks. The basic unit in argument structure analysis tasks is the argument component, a text span that can express a complete opinion. Compared with entities or events in traditional information extraction tasks (Yan et al. 2021b; Lu et al. 2022), argument components are longer and have more ambiguous boundaries, making them more challenging to identify. Additionally, the granularity of argument components defined by different argument structure analysis tasks varies, making it difficult to efficiently handle them with simple prompt designs. Therefore, we believe designing a unified generative framework for argument structure analysis tasks is more challenging. To the best of our knowledge, we are the first to attempt a unified solution for different argument structure analysis tasks. Compared with the unified generative frameworks for information extraction (Yan et al. 2021b; Lu et al. 2022) or sentiment analysis (Yan et al. 2021a; Li et al. 2023), we propose a multi-view learning strategy specifically designed for the characteristics of argument structure analysis tasks, which further enhances UniASA's performance.

3. Methodology

Our framework, named UniASA, is illustrated in Figure 4. UniASA leverages a unified task template to consolidate the input and output format of various argument structure analysis tasks. In this way, it casts these divergent tasks into a sequence-to-sequence generation task. Then, UniASA uses a pre-trained generative model to address this generation task. Further, UniASA is enhanced by a multi-view learning strategy, in which we consider subtasks decoupled from the original argument structure analysis task as additional perspectives for learning. Each of these subtasks, along with the original task, are modeled as distinct views.

3.1 Unified Task Template

We design a unified task template to transform various argument structure analysis tasks into a consistent format of natural language input-output pairs. These pairs can



[ASA] Please do an argument structure analysis task.
Topic: Cloning is beneficial.
Text:
Cloning can benefit those needing organ transplants. Cloned organs can exactly match patients' blood and tissue types since they can be raised from cloned stem cells of the patient.

The type of [# Cloning can benefit those needing organ transplants #] is "Claim:For". It produces no relation.
[SEP]
The type of [# Cloned organs can exactly match patients' blood and tissue types #] is "Premise". It is a "Support" type of argument for [# Cloning can benefit those needing organ transplants #] ("Claim:For").
[SEP]
The type of [# they can be raised from cloned stem cells of the patient #] is "Premise". It is a "Support" type of argument for [# Cloned organs can exactly match patients' blood and tissue types #] ("Premise").
[SEP]

[ASA] Please do an argument structure analysis task.
Topic: None
Text:
[Review]:
[A] **[# 1 #]** This paper studies the … [/A]
[A] **[# 2 #]** This work can only be seen as … [/A]
[A] **[# 3 #]** Most of the merge functions … [/A]
[A] **[# 4 #]** Moreover, the sample … [/A]
[Rebuttal]:
[A] **[# 5 #]** Thank you for your careful … [/A]
[A] **[# 6 #]** At Section 2 we discuss how … [/A]
[A] **[# 7 #]** The motivating setting for our … [/A]

The type of [# 2 3 #] is "Argument". It can form an "Argument-Pair" with [# 6 #].
[SEP]
The type of [# 4 #] is "Argument". It can form an "Argument-Pair" with [# 7 #].
[SEP]
The type of [# 6 #] is "Argument". It produces no relation.
[SEP]
The type of [# 7 #] is "Argument". It produces no relation.

[ASA] Please do an argument structure analysis task.
Topic: Should vaccination be mandatory?
Text:
[A] **[# 1 #]** Vaccines can save children's … [/A]
[A] **[# 2 #]** The American Academy of … [/A]
[A] **[# 3 #]** According to Shot@Life, … [/A]

The type of [# 1 #] is "Support". It is supported by the "Expert" evidence [# 2 #], and is supported by the "Expert" evidence [# 3 #].
[SEP]

Figure 4
The overall framework of UniASA. It utilizes the source sequence template and target sequence template to convert the input and output of each argument structure analysis task into a source-target sequence pair. This figure specifically demonstrates how the three examples illustrated in Figures 1, 2, and 3 can be converted into source-target sequence pairs by our templates. Notably, for simplicity, we shorten the source sequences for the APE and AQE task examples. Also, in the APE task, since all argumentative relations are bidirectional, we only need to extract the relations in one direction. Here, we only consider the argumentative relations from the review to the rebuttal. "[ASA]" is a special token used to mark different views in multi-view learning, as detailed in Section 3.3. "[A]" and "[/A]" are special tokens used to mark each sentence. "[# i #]" indicates the index of each sentence. The underlines in the target sequence mark the prediction targets of each task: argument components, component types, and argumentative relation types.

naturally be viewed as the source-target sequence pairs in a sequence-to-sequence generation task. Therefore, any generative language model could be used to learn and solve different tasks in a unified sequence-to-sequence manner. Specifically, the unified task template consists of two parts: the *source sequence template* that transforms task instruction, input topic, and input text into a source sequence, and the *target sequence template* that specifies the structural output format. Three examples, each for the E2E-AM, APE, and AQE tasks, are illustrated in Figure 4.

3.1.1 Source Sequence Template. The source sequence template consolidates the task instruction, the input topic, and the input text of various argument structure analysis tasks in a consistent source sequence. The descriptions of each element are as follows:

Task Instruction. It describes the objective of argument structure analysis tasks. Accordingly, we standardize the task instruction of all tasks to be: *"[ASA] Please do an argument structure analysis task."*

Input Topic. Since debates are usually centered around a specific topic, knowing the topic of argumentation is crucial for argument structure analysis. The AAEC and QAM datasets label each document sample with the corresponding topic.[2] Parts of the E2E-AM and APE tasks in Figure 4 show examples of topic information in the AAEC and QAM datasets. For other datasets that do not contain topics, we indicate the input topic to be "None".

Input Text. It is the input argumentative document. Different argument structure analysis datasets correspond to different document types, which could be student essays, online debate comments, academic papers, and so on. For the E2E-AM task, we directly use the unaltered document as input text. For the AQE task, given that its argument component granularity is at the sentence level, we need to alter all sentences in the input document to include their sentence indexes. In practice, we follow Guo et al. (2023) to insert the sentence index "[# i #]" into the start of each sentence. We also use two special tokens, "[A]" and "[/A]" to wrap each sentence along with its sentence index, as shown in Figure 4. In this way, in the target sequence, sentence-level argument components can be concisely and efficiently represented by their sentence indexes. For the APE task, which also identifies sentence-level components, we adopt the same method as the AQE task to add an index in front of each sentence. Additionally, since the input of the APE task includes two documents, namely, the review and rebuttal, we prepend each with special tokens "[Review]" and "[Rebuttal]", respectively, and then concatenate them to form the input text.

Finally, the task instruction, input topic, and input text are concatenated to be our source sequence, SS_{asa}.

3.1.2 Target Sequence Template. The objective of all argument structure analysis tasks is to achieve two subtasks, that is, argument component identification (ACI) and argumentative relation identification (ARI). Therefore, our target sequence template consists of two types of subsequences: the ACI subsequences S_{aci} and the ARI subsequences S_{ari}.

[2] In the experiments, to fairly compare with previous state-of-the-art methods on the AAEC and QAM datasets (ST [Morio et al. 2022] and QuadTAG [Guo et al. 2023]), we adopt the same experimental setup as them, assuming that the topic information is known during both training and testing.

Table 2
The task-specific template $\mathcal{M}(\cdot)$ for different argument structure analysis tasks.

Task	Template $\mathcal{M}(\cdot)$
E2E-AM	"is a/an '$\mathcal{T}_r(\cdot)$' type of argument for"
APE	"can form an 'Argument-Pair' with"
AQE	"is supported by the '$\mathcal{T}_r(\cdot)$' evidence"

Each subsequence corresponds to an argument component, with the ACI subsequence and ARI subsequence describing its type and associated argumentative relations.

ACI Subsequence S_{aci}. As depicted in Table 1, the definitions of argument component granularity and categories vary across different argument structure analysis tasks. To tackle this challenge, we devise a template to uniformly represent the expected output of the ACI subtask for each argument structure analysis task. Specifically, S_{aci} can be expressed as:

$$S_{aci}(a_i) = \text{"The type of } [\# \, a_i \, \#] \text{ is '}\mathcal{T}_c(a_i)\text{'."}$$

where a_i represents the i-th argument component from the input document. In the E2E-AM task, it is a text span, whereas in the APE task, it is one or more sentence indexes, and in the AQE task, it is a single sentence index. $\mathcal{T}_c(a_i)$ denotes the type of a_i. The category set for $\mathcal{T}_c(a_i)$ differs across various datasets, as shown in the "Component Category" column of Table 1. For example, in the AAEC dataset, $\mathcal{T}_c(a_i) \in$ {MajorClaim, Claim:For, Claim:Against, Premise}.[3]

ARI Subsequence S_{ari}. For identifying argumentative relations, we design the following S_{ari} for each argument component to express all of its associated relations:

$$S_{ari}(a_i) = \text{"}\textbf{It } \mathcal{M}(\mathcal{T}_r(a_i, a_j)) \, [\# \, a_j \, \#] \, (\text{'}\mathcal{T}_c(a_j)\text{'}), \textbf{ and } \mathcal{M}(\mathcal{T}_r(a_i, a_{j+1})) \, [\# \, a_{j+1} \, \#]$$
$$(\text{'}\mathcal{T}_c(a_{j+1})\text{'}), \textbf{and} \ldots \text{"}$$

where $a_j, a_{j+1}, \ldots$ represent the argument components related to a_i.[4] $\mathcal{T}_r(a_i, a_j)$ is the type of the argumentative relation between a_i and a_j. Similarly to $\mathcal{T}_c(\cdot)$, the category set of $\mathcal{T}_r(\cdot)$ is also dataset-specific, as shown in the "Relation Category" column of Table 1. $\mathcal{M}(\cdot)$ is a set of templates designed to translate $\mathcal{T}_r(a_i, a_j)$ into a natural language sequence. $\mathcal{M}(\cdot)$ is task-specific for different argument structure analysis tasks, as demonstrated in Table 2. These task-specific templates are necessary due to the differing definitions of argumentative relations across different argument structure analysis tasks. For instance, argumentative relations are unidirectional, mostly representing support or negation from one argument component to another. In the APE task, however, relations are undirected, indicating a discussion about the same issue between two argument components. Therefore, to achieve unified modeling, it is essential to tailor distinct templates for the argumentative relations in each argument structure analysis task.

[3] Since the APE task does not define argument component categories, we simply set $\mathcal{T}_c(a_i)$ as "Argument".
[4] If there is no relation from a_i to any other component, we set $S_{ari}(a_i)$ as "It produces no relation".

For each argument component a_i, we concatenate $S_{aci}(a_i)$ and $S_{ari}(a_i)$ to form a complete subsequence $S(a_i) = S_{aci}(a_i) \oplus S_{ari}(a_i)$. Then, all the complete subsequences are (i) joined together according to their corresponding components' order of appearance in the input document, and (ii) separated by "[SEP]", to form the final target sequence:

$$\text{TS}_{asa} = \text{``}S(a_1)\,\text{[SEP]}\,S(a_2)\,\text{[SEP]}\dots\text{''}$$

3.2 Generative Framework

Once the source and target sequences are prepared, we can fine-tune a pre-trained sequence-to-sequence generation model, like T5, with the standard negative log-likelihood loss. For inference, we just need to convert the inputs of different argument structure analysis tasks into a standardized source sequence (SS_{asa}), and feed it into the fine-tuned model for target sequence generation. The argument structure can be easily decoded from the generated target sequence.

3.3 Multi-view Learning

Furthermore, we argue that this simple generative framework can be further enhanced through a multi-view learning strategy. To elaborate, each argument structure analysis task can be decomposed into two subtasks: argument component identification (ACI) and argumentative relation identification (ARI). These two subtasks can be considered as two distinct views (the ACI view and the ARI view), while solving the argument structure analysis task end-to-end, as described in Section 3.1, can be regarded as a third view (the ASA view). Accordingly, we suggest leveraging the instruction-following capabilities of pre-trained generative models to learn from these three views simultaneously. More precisely, for each data sample, we not only construct an end-to-end source-target sequence pair for the ASA view (Section 3.1), but also develop two additional pairs for the ACI and ARI views. Figure 5 illustrates our multi-view learning strategy. Also, a specific example is shown in Appendix I.

3.3.1 ACI View. For the source sequence of the ACI view SS_{aci}, we simply change the task instruction in SS_{asa} to "*[ACI] Please do an argument component identification task*", with all other parts remaining the same as described in Section 3.1.1.

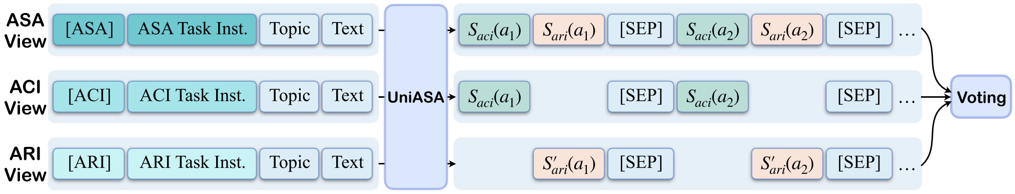


Figure 5
Multi-view learning. "Inst." stands for "Instruction". "Topic" and "Text" refer to the "Input Topic" and "Input Text" described in Section 3.1.1. A specific example is shown in Appendix I.

For the target sequence TS_{aci}, we just join all the ACI subsequences $S_{aci}(a_i)$ together, and separate them by "[SEP]":

$$\mathrm{TS}_{aci} = \text{“}S_{aci}(a_1)\,\text{[SEP]}\,S_{aci}(a_2)\,\text{[SEP]}\ldots\text{”}$$

3.3.2 ARI View. Similar to the ACI view, for the source sequence of the ARI view SS_{ari}, we only change the task instruction in SS_{asa} to *"[ARI] Please do an argumentative relation extraction task"*.

For the target sequence TS_{ari}, we make simple modifications to S_{ari} to meet the requirements of the ARI view. Specifically, first, "It" in S_{ari} is replaced with "[# a_i #]" as we need to first identify the source component in an argumentative relation. Secondly, the elements related to component types in S_{ari}, like "('$\mathcal{T}_c(a_j)$')", are omitted, as we want the ARI view to focus solely on argumentative relations. In this way, we can obtain the modified $S'_{ari}(a_i)$:

$$S'_{ari}(a_i) = \text{“[\# } a_i \text{ \#]}\, \mathcal{M}(\mathcal{T}_r(a_i, a_j))\, \text{[\# } a_j \text{ \#]},\ \textbf{and}\ \mathcal{M}(\mathcal{T}_r(a_i, a_{j+1}))\, \text{[\# } a_{j+1} \text{ \#]},\ \textbf{and}\ldots\text{”}$$

Then, all the modified ARI subsequences $S'_{ari}(a_i)$ are joined together, and separated by "[SEP]". The resulting target sequence for the ARI view is:

$$\mathrm{TS}_{ari} = \text{“}S'_{ari}(a_1)\,\text{[SEP]}\,S'_{ari}(a_2)\,\text{[SEP]}\ldots\text{”}$$

3.3.3 Multi-view Training and Inference. For each data sample, we can obtain three source-target pairs, namely, <SS_{asa}, TS_{asa}>, <SS_{aci}, TS_{aci}>, and <SS_{ari}, TS_{ari}>, for the ASA, ACI, and ARI views, respectively. During training, the three types of pairs of all data samples from the training set are merged together to fine-tune a pre-trained generation model, achieving multi-view training.

During inference, the model will generate three predicted target sequences for each test data sample. From the predicted TS_{asa}, we can obtain predictions for both the ACI and ARI subtasks together. Meanwhile, from the predicted TS_{aci} and TS_{ari}, we can get the predictions for the ACI and ARI subtasks separately. In other words, multi-view inference provides two predictions from two different views for both the ACI and ARI subtasks. Consequently, based on the two predictions for the ACI/ARI subtasks, we adopt a simple voting strategy to determine the final prediction. That is, we retain the prediction only if it is consistent across both views. This ensures that only predictions with high confidence are kept in the final predictions. In addition, it should be noted that results with inconsistent votes across different views are discarded.

4. Experiments

In this section, we detail the tasks and datasets used in our experiments, the implementation details, and the evaluation metrics. We group the baseline methods by the specific tasks these methods were designed for, including the E2E-AM, APE, and AQE tasks.

4.1 Tasks and Datasets

We test UniASA on seven datasets over three tasks, including the end-to-end E2E-AM, AQE, and APE tasks. The definitions and objectives of each task are detailed in Table 1. For the E2E-AM task, we conduct experiments on five datasets: AAEC (Stab and Gurevych 2017), CDCP (Park and Cardie 2018), AbstRCT (Mayer, Cabrio, and Villata

2020), MTC (Peldszus and Stede 2015), and AASD (Accuosto, Neves, and Saggion 2021). Note that our experiments on the AAEC dataset are, by default, conducted on essay-level data. However, for a fair comparison with previous methods (Eger, Daxenberger, and Gurevych 2017; Ye and Teufel 2021; Bao et al. 2022a), we also perform experiments on the paragraph-level data of the AAEC dataset. Additionally, following Morio et al. (2022), we also carry out experiments on the five E2E-AM datasets with oracle argument component spans. In this setup, we mark the component spans in the "Input Text" using the component index token "[# i #]" along with special tokens "[A]" and "[/A]", similar to how we handle the AQE task in Section 3.1.1. This allows us to predict only the component index in the target sequence, instead of the component span. For the AQE task, we adopt the QAM dataset (Guo et al. 2023). For the APE task, we perform experiments on the RR-Submission-v2 (RR) dataset (Cheng et al. 2021; Sun et al. 2023).

The statistics of these datasets are shown in Appendix K. For fair comparisons, we follow the data splits used in previous studies (Morio et al. 2022; Guo et al. 2023; Cheng et al. 2021). For the MTC and AASD datasets, we follow the work of Morio et al. (2022) and conduct experiments using five-fold cross-validation.

For the selection of the aforementioned datasets, our principle is to use the most widely recognized and mainstream datasets that have been utilized in recent argument structure analysis studies. This ensures appropriate comparison and analysis with existing approaches. Therefore, we investigate the most frequently used argument structure analysis datasets in top NLP conferences and journals over the past five years. Ultimately, we select these seven most mainstream datasets.

4.2 Implementation Details

Given the long input texts for the argument structure analysis tasks, we adopt LongT5 (Guo et al. 2022) as our base model. LongT5 is an extension of T5 (Raffel et al. 2020), which can utilize efficient attention mechanisms to boost its ability for processing long sequences. To be specific, we use the pre-trained long-t5-tglobal-base[5] model with the transient-global attention mechanism. Additionally, for the QAM dataset, we utilize the original t5-base[6] to align with Guo et al. (2023). For the CDCP, MTC, and AASD datasets, we use t5-base due to the shorter length of input texts. We also provide a detailed discussion of potential data leakage issues when using the T5 models in Appendix C.

During training, we use Adam and a linear scheduler with a warm-up phase. Since Morio et al. (2022) perform extensive hyperparameter tuning, we also conduct a hyperparameter search for a fair comparison. Specifically, we search for the learning rate, weight decay, epoch number, warm-up ratio, and batch size on the validation set. The tuned hyperparameters for the main experiments are displayed in Table 3. The optimal checkpoints are selected according to the average score of the component identification task and the relation identification task on the validation set. We run all experiments 10 times and report the mean scores. Also, when comparing UniASA to the current SOTA methods in our main experiments, we conduct one-sample t-tests for statistical significance testing. During inference, we use greedy decoding. Since the generated text may not always precisely match the input text, we use fuzzysearch[7] to align the model-generated text with the corresponding position of the input text. For

[5] `https://huggingface.co/google/long-t5-tglobal-base`.
[6] `https://huggingface.co/t5-base`.
[7] `https://github.com/taleinat/fuzzysearch`.

Table 3
Hyperparameters for the main experiments.

Task	Dataset	Learning Rate	Batch Size	Epoch	Warm-up Ratio	Weight Decay
E2E-AM	AAEC	1e-4	1	35	0.0	0.0
	CDCP	2e-4	2	40	0.01	0.1
	AbstRCT	1e-4	2	10	0.0	0.0
	MTC	1e-4	2	50	0.1	0.0
	AASD	3e-4	2	100	0.1	0.0
APE	RR	1e-4	1	10	0.1	0.1
AQE	QAM	1e-4	2	8	0.1	0.1

the intermediate fine-tuning experiments, the models are pre-trained 10 epochs, and the learning rate and batch size are set to 1e-5 and 2. The warm-up ratio and weight decay are set to 0.0. All fine-tuning experiments are conducted on a single NVIDIA H100 GPU.

4.3 Evaluation Metrics

For the E2E-AM task, we utilize both macro and micro F1 scores as the main metrics for the ACI ($F1_{aci}$) and the ARI ($F1_{ari}$) subtasks, in line with prior studies (Stab and Gurevych 2014; Morio et al. 2022). For $F1_{aci}$, a predicted argument component is regarded as correct if and only if its span and type are the same as the gold component. For $F1_{ari}$, a predicted argumentative relation is considered as correct if and only if its source component span, target component span, and relation type are identical to those of the gold relation. Note that when calculating $F1_{ari}$, the component type is not considered. Hence, we introduce a more comprehensive evaluation metric, argumentative relation identification with component type, $F1_{arict}$ (micro), where a predicted argumentative relation is only deemed correct if its source component span, source component type, target component span, target component type, and relation type are exactly the same as the gold relation. Furthermore, following the work of Morio et al. (2022), we also adopt $F1_{span}$ and $F1_{link}$. $F1_{span}$ is a variation of micro $F1_{aci}$ without considering the component type. $F1_{link}$ is a micro $F1_{ari}$ variation that ignores the relation type.

For the APE task, following previous work (Cheng et al. 2021), we use F1 scores to evaluate the argument component identification task ($F1_{aci}$)[8] and the argument pair extraction task ($F1_{ape}$). Here, according to the definition of the APE task, the $F1_{aci}$ focuses solely on the recognition of sentence-level argument component spans, excluding their types. Likewise, the $F1_{ape}$ only considers the existence of relations between components, disregarding the types of these relations.

For the AQE task, we report F1 scores ($F1_{aqe}$) on both the validation and the test sets following Guo et al. (2023). Here, a predicted quadruplet is considered correct only if all its four elements match the gold quadruplet exactly, that is, (i) the sentence index of the source component (claim), (ii) the stance of the source component, (iii) the sentence index of the target component (evidence), and (iv) the type of evidence for the target component. Additionally, for the APE and AQE tasks, we report the precision and recall scores as done in previous work (Cheng et al. 2021; Guo et al. 2023).

[8] In previous work, this subtask is termed "argument mining (AM)" in the APE task. To avoid confusion with the E2E-AM task in this article, we refer to it as "argument component identification (ACI)".

4.4 Compared Methods

For the E2E-AM task, we compare UniASA against the following methods:

- **BLCC** (Eger, Daxenberger, and Gurevych 2017) solves the E2E-AM task by sequence labeling.
- **LSTM-ER** (Eger, Daxenberger, and Gurevych 2017) is an end-to-end relation extraction approach with a tree-structured and sequential LSTM.
- **BiPAM-syn** (Ye and Teufel 2021) is a dependency parsing model based on a biaffine network, which is augmented by syntactic information.
- **ST** (Morio et al. 2022) is a span-biaffine framework based on Longformer.
- **GMAM** (Bao et al. 2022a) is a generative approach with a constrained pointer mechanism.

For the E2E-AM task with oracle argument component spans, we compare the following methods:

- **Joint-ILP** (Stab and Gurevych 2017) jointly classifies component types and extracts relations by integer linear programming.
- **Ptr. Net.** (Potash, Romanov, and Rumshisky 2017) adapts a pointer network for recognizing argumentative relations.
- **Span-LSTM** (Kuribayashi et al. 2019) applies the LSTM-minus-based span representation to solve the E2E-AM task.
- **BERT-Trans** (Bao et al. 2021a) is a transition-based method that parses an argument graph by a series of actions.
- **TSP-PLBA** (Morio et al. 2020) is a biaffine attention-based parser for non-tree argument structure.
- **Rel.RoBERTa/SciBERT** (Mayer, Cabrio, and Villata 2020) is designed for the E2E-AM task in clinical texts, built upon RoBERTa (Liu et al. 2019b) or SciBERT (Beltagy, Lo, and Cohan 2019).

For the APE task, we adopt the following strong baselines for comparison:

- **MLMC** (Cheng et al. 2021) is a table-filling-based model.
- **MGF** (Bao et al. 2021b) addresses this task through a mutual guided framework.
- **MRC-APE** (Bao et al. 2022b) transforms the APE task into a multi-turn reading comprehension task.
- **MMR-GCN** (Zhu et al. 2023) proposes a mutually enhanced model with a multi-scale relation-aware graph.
- **PIGEON** (Sun et al. 2023) tackles the APE task with probing graph decomposition.

For the AQE task, we compare the following methods:

- **T5-Struc.** (Guo et al. 2023) applies T5-base to solve the AQE task.
- **QuadTAG** (Guo et al. 2023) enhances T5 by introducing a tagging table.

5. Main Experimental Results

We discuss the results of UniASA against the baselines designed for each argument structure analysis task. The results for the E2E-AM task with oracle spans are in Appendix A. The detailed ablation analysis of multi-view learning can be found in Appendix J.

5.1 Results for the E2E-AM Task

The experimental results for the E2E-AM task over five datasets are shown in Table 4. Also, we provide detailed F1 scores for each type of component and relation in Appendix F. Overall, UniASA outperforms the best baseline, ST, on the AAEC, AbstRCT, and MTC datasets, while achieving comparable performance on the CDCP and AASD datasets. The removal of multi-view learning (*w/o mv*) decreases the overall performance of UniASA, but the results are still satisfactory compared to ST. Additionally, on the AAEC and AbstRCT datasets, UniASA shows more significant improvements over ST in F1$_{arict}$ than in F1$_{aci}$ and F1$_{ari}$. This suggests that our method more effectively models the interconnection between the two subtasks, ACI and ARI, and better ensures their consistency during prediction, thus achieving superior F1$_{arict}$. The reason might be that UniASA transforms the prediction targets of the two subtasks into one target text sequence. This sequence, like natural language text, is self-consistent. Therefore, generative models pre-trained on natural language texts can achieve high consistency between the two subtasks. Below, we specifically discuss the results on each dataset.

AAEC. On the AAEC dataset, our method outperforms ST on both essay-level data and paragraph-level data. Further, by comparing the improvements in Micro F1$_{aci}$ and Micro F1$_{ari}$, we observe that the improvements are more pronounced in essay-level data than in paragraph-level data. This highlights that our model is more effective in handling longer texts. It is noteworthy that UniASA's Macro F1$_{ari}$ score is slightly lower than that of ST. This suggests that our model may have limited capability in learning imbalanced relation types compared to ST. This is primarily because the majority of argumentative relations in the AAEC dataset are "Support", while "Attack" relations are quite scarce (Bao et al. 2021a). More analysis on this issue is in Appendix F.

CDCP. According to Table 4, for the CDCP dataset, UniASA underperforms ST on most metrics, with the main gap arising from ARI-related metrics. This might be due to the lower relation density in the CDCP dataset (Morio et al. 2022), namely, the ratio of the argumentative relation numbers to the component numbers in each sample. The low relation density suggests a higher degree of data imbalance, where the total number of argumentative relations is quite limited, and most component pairs have no relationship. We hypothesize that ST's method of classifying the relation of each component pair through pairwise matching may be better suited for handling such an imbalanced dataset. In contrast, UniASA simply generates argumentative relations in a sequence without directly addressing non-related component pairs, which may not be

Table 4
Main experimental results for the E2E-AM task. The F1$_{arict}$ score of ST is calculated based on reproduced results from the official code provided by Morio et al. (2022). The best scores are in boldface. *"para."* indicates *"paragraph-level"*. *"w/o mv"* denotes the results after removing multi-view learning from UniASA. * represents statistical significance with $p < 0.05$. $^{\diamond}$ indicates the current SOTA method. † denotes that the scores are computed by the evaluation metric from the work of Eger, Daxenberger, and Gurevych (2017).[9]

Dataset	**Method**	F1$_{span}$	F1$_{aci}$		F1$_{link}$	F1$_{ari}$		F1$_{arict}$
			Micro	Macro		Micro	Macro	
AAEC	ST$^{\diamond}$	85.21	75.54	66.59	55.66	55.17	**42.30**	46.57
	UniASA	86.56*	**77.60***	**67.81***	**57.22***	**56.33***	41.62	**49.83***
	w/o mv	**86.84**	76.24	65.75	56.49	55.37	39.54	48.59*
	BLCC†	–	63.23	–	–	34.82	–	–
	LSTM-ER†	–	66.21	–	–	29.56	–	–
	ST$^{\dagger\diamond}$	–	76.55	–	–	54.66	–	51.69
	UniASA†	86.56	**78.40***	75.29	56.87	**55.49**	47.73	**52.85***
	w/o mv†	86.84	77.24	73.79	55.76	54.08	44.78	51.54
AAEC (*para.*)	BLCC†	–	66.69	–	–	39.83	–	–
	LSTM-ER†	–	70.83	–	–	45.52	–	–
	BiPAM-syn†	–	73.50	–	–	46.40	–	–
	GMAM†	–	75.94		–	–	–	50.08
	ST$^{\dagger\diamond}$	–	76.48	–	–	59.55	–	53.78
	UniASA†	85.50	**76.80**	73.44	60.59	**59.86**	50.54	**55.65***
	w/o mv†	86.37	76.53	72.75	60.89	58.27	50.25	55.25*
CDCP	ST$^{\diamond}$	**82.88**	**68.90**	65.78	**31.94**	**31.94**	**16.26**	**24.96**
	UniASA	82.47	68.16	**72.31***	30.57	30.57	15.56	23.29
	w/o mv	81.45	67.64	68.57*	28.37	28.37	14.46	21.86
AbstRCT	ST$^{\diamond}$	70.29	64.16	45.04	39.35	38.38	**31.91**	35.21
	UniASA	**74.35***	**69.67***	**48.19***	**40.45***	**39.11**	30.62	**38.65***
	w/o mv	73.62*	68.56*	45.37	36.84	35.95	27.52	34.84
MTC	ST$^{\diamond}$	**87.68**	78.83	**73.77**	**53.43**	45.92	33.07	43.03
	UniASA	87.26	**79.88**	73.61	52.35	**46.76***	**34.66**	**44.44***
	w/o mv	86.24	78.90	71.70	50.76	45.68	33.06	42.68
AASD	ST$^{\diamond}$	**87.10**	**69.06**	**58.06**	**54.82**	**49.83**	**42.10**	**44.86**
	UniASA	82.89	68.84	57.10	54.58	49.54	41.40	44.78
	w/o mv	82.67	68.17	55.61	52.44	48.23	40.65	42.32

as effective as ST in such a dataset. Fortunately, most datasets, like AAEC and AbstRCT, have a higher degree of relation density, and UniASA is more adept at handling them.

AbstRCT. Regarding the AbstRCT dataset, UniASA significantly surpasses ST in metrics related to the ACI subtask, and noticeably outperforms ST in most ARI-related metrics. However, UniASA's Macro F1$_{ari}$ is lower than ST's. Similar to our observations in the AAEC and CDCP datasets, UniASA shows a marginally weaker capability in handling imbalanced data than ST.

MTC and AASD. On most metrics, UniASA achieves results similar to or better than ST. Both MTC and AASD are relatively small datasets, with training sets containing fewer than 100 instances each. The fact that UniASA can achieve such results on these small datasets demonstrates its adaptability in low-resource conditions.

[9] This metric is slightly different from our main evaluation metric for the E2E-AM task (described in Section 4.3). Specifically, for the AAEC dataset, (1) "Claim:Against" and "Claim:For" are uniformly treated as "Claim". (2) Each "Claim" is assumed to be linked only to the last "MajorClaim" in input text, with the argumentative relation type being "Against" or "For". (3) Each "MajorClaim" is assumed to have a "None" type relation with a pseudo root node. For more details, please refer to this link.

Table 5
Main experimental results for the APE task. * denotes statistical significance with $p < 0.05$. $^\diamond$ indicates the current SOTA method.

Dataset	Method	P_{aci}	R_{aci}	$F1_{aci}$	P_{ape}	R_{ape}	$F1_{ape}$
RR	MLMC	69.53	73.27	71.35	37.15	29.38	32.81
	MGF	70.82	73.19	71.99	40.45	30.77	34.95
	MRC-APE	71.83	73.05	72.43	41.83	38.17	39.92
	MMR-GCN	71.68	70.67	71.16	**44.69**	38.44	41.31
	PIGEON$^\diamond$	72.29	73.22	72.75	41.06	**44.84**	42.86
	UniASA	**75.04***	73.19	**74.10***	44.42*	41.78	43.05
	w/o mv	74.12*	**73.41**	73.77*	44.14*	42.98	**43.53***

5.2 Results for the APE Task

According to Table 5, on the RR dataset, UniASA outperforms all baselines in terms of $F1_{aci}$ and $F1_{ape}$ scores. In particular, since UniASA determines final predictions by voting, its advantage in precision scores is especially significant. Additionally, by comparing UniASA to "UniASA *w/o mv*", we find that multi-view learning does not enhance UniASA and even negatively impacts its $F1_{ape}$ score. We argue that since the APE task does not involve the classification of components and relations like the E2E-AM and AQE tasks, applying multi-view learning based on subtask decomposition to the APE task is somewhat unnatural. Therefore, these results are not surprising. Even so, it is evident that UniASA without multi-view learning (*w/o mv*) achieves a significantly higher $F1_{ape}$ than PIGEON, demonstrating the advantage of using our generative framework for the APE task. Also, to our knowledge, we are the first to explore solving the APE task with a generative approach.

5.3 Results for the AQE Task

As indicated in Table 6, our UniASA significantly surpasses the current SOTA method, QuadTAG, on both the validation and test sets of the QAM dataset. Although UniASA's recall scores are similar to those of QuadTAG, its precision scores are substantially higher, highlighting the effectiveness of our multi-view learning and voting strategies. Also, the performance degradation after eliminating multi-view learning (*w/o mv*) demonstrates the necessity of multi-view learning for the AQE task. By comparing

Table 6
Main experimental results for the AQE task. * denotes statistical significance with $p < 0.05$. $^\diamond$ indicates the current SOTA method.

Dataset	Method	Validation			Test		
		P_{aqe}	R_{aqe}	$F1_{aqe}$	P_{aqe}	R_{aqe}	$F1_{aqe}$
QAM	T5-Struc.	17.14	16.60	16.87	21.16	18.16	19.54
	QuadTAG$^\diamond$	20.55	18.82	19.64	24.47	19.01	21.39
	UniASA	**23.64***	18.58	**20.81***	**29.13***	19.73	**23.52***
	w/o mv	20.75	**20.07***	20.40	24.69	**20.20***	22.22

"UniASA *w/o mv*" with T5-Struc. (Guo et al. 2023), we find that although both are based on simple fine-tuning of T5, our method significantly outperforms T5-Struc. This may be because our designed target sequence template is more akin to natural language, making it more suitable for fine-tuning the generative pre-trained model, T5.

5.4 Results of Binning Test Data Samples by the Number of Gold Components

In Appendix G, we analyze model performance by binning test data samples by the number of gold components contained in each sample. As a result, we divide the original test set into three bins, each corresponding to data samples containing high, medium, and low quantities of gold components. To summarize, on the E2E-AM task, UniASA demonstrates particularly superior performance compared to ST for data samples with high quantities of gold components. On the APE and AQE tasks, UniASA is more adept than the baseline models at handling data samples with low or medium quantities of gold components. For detailed analysis, please refer to Appendix G.

6. Further Analysis

6.1 Results of Fine-tuning Larger Language Models

UniASA is a versatile generative framework that can be directly adapted to fine-tune various larger language models, thereby achieving better performance. Here, we present the results of fine-tuning t5-large,[10] t5-3b,[11] and Llama-3-8B[12] (Touvron et al. 2023) in Table 7. For implementation details, please refer to Appendix B. Overall, the performance of UniASA consistently improves as the number of model parameters increases. Additionally, the decoder-only Llama-3-8B model achieved the best overall results. These findings demonstrate the strong adaptability of UniASA to pre-trained models of varying sizes and architectures.

6.2 Results of In-context Learning

In addition to fine-tuning, the UniASA framework can also be applied to larger-scale models through in-context learning. Here, we conduct few-shot in-context learning experiments with "UniASA *w/o mv*" using ChatGPT (GPT-4o), Llama-3-70B-Instruct, and Mixtral8×7B-Instruct-v0.1. The implementation details and prompt templates for these experiments can be found in Appendix E. Note that the multi-view learning strategy is not incorporated due to its negligible contribution in our preliminary experiments.

The results of GPT-4o are presented in Table 8. The results of Llama-3-70B-Instruct and Mixtral-8×7B-Instruct-v0.1 are shown in Appendix D. Overall, large language models can understand the task instructions through in-context learning, and learn the argument structure analysis task to a certain degree. For GPT-4o, we observe a general trend of improved performance as we increase the number of examples. However, the performance gains become notably less pronounced when scaling from 16 to 32 examples, with some metrics actually showing slight decreases. Therefore, we believe that further increasing the number of examples would likely yield limited

[10] `https://huggingface.co/google-t5/t5-large`.
[11] `https://huggingface.co/google-t5/t5-3b`.
[12] `https://huggingface.co/meta-llama/Meta-Llama-3-8B`.

Table 7
Fine-tuning results of UniASA with larger language models. Note that the multi-view learning strategy is incorporated during fine-tuning.

Task	Dataset	Model	# Parameters	F1_{aci} Micro	F1_{ari} Micro	F1_{arict}	F1_{ape}	F1_{aqe} Test
E2E-AM	AAEC	long-t5-tglobal-base	247M	77.60	56.33	49.83	–	–
		t5-large	770M	77.71	58.09	50.81	–	–
		t5-3b	3B	78.67	57.43	51.49	–	–
		llama-3-8b	8B	**79.70**	**59.34**	**55.27**	–	–
	CDCP	t5-base	220M	68.16	30.57	23.29	–	–
		t5-large	770M	69.63	30.23	23.59	–	–
		t5-3b	3B	69.09	32.23	24.26	–	–
		llama-3-8b	8B	**70.83**	**35.58**	**28.69**	–	–
	AbstRCT	long-t5-tglobal-base	247M	69.67	39.11	38.65	–	–
		t5-large	770M	73.50	40.11	38.41	–	–
		t5-3b	3B	**74.70**	40.37	38.82	–	–
		llama-3-8b	8B	74.14	**52.22**	**49.13**	–	–
	MTC	t5-base	220M	79.88	46.76	44.44	–	–
		t5-large	770M	78.88	46.58	45.20	–	–
		t5-3b	3B	**83.53**	49.90	48.62	–	–
		llama-3-8b	8B	82.26	**57.52**	**55.87**	–	–
	AASD	t5-base	220M	68.84	49.54	44.78	–	–
		t5-large	770M	67.57	49.78	40.45	–	–
		t5-3b	3B	70.38	50.39	43.12	–	–
		llama-3-8b	8B	**75.71**	**59.06**	**52.11**	–	–
APE	RR	long-t5-tglobal-base	247M	74.10	–	–	43.05	–
		t5-large	770M	74.90	–	–	46.19	–
		t5-3b	3B	75.17	–	–	51.84	–
		llama-3-8b	8B	**75.56**	–	–	**53.70**	–
AQE	QAM	t5-base	220M	–	–	–	–	23.52
		t5-large	770M	–	–	–	–	27.88
		t5-3b	3B	–	–	–	–	30.07
		llama-3-8b	8B	–	–	–	–	**30.24**

performance improvements. Moreover, compared to the open-source Llama-3-70B-Instruct and Mixtral-8×7B-Instruct-v0.1, the proprietary GPT-4o has significant advantages in both context window size (128k) and task handling capability. Although Llama-3-70B-Instruct performs similarly to GPT-4o on the E2E-AM task in the 4-shot setting, its shorter context window size (8k) limits its capabilities, allowing experiments only up to the 4-shot setting. While Mixtral-8×7B-Instruct-v0.1 has a longer context window size (32k), its overall performance is noticeably inferior to Llama-3-70B-Instruct and GPT-4o.

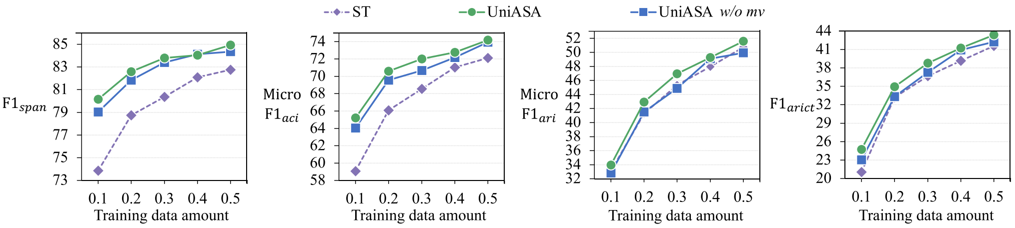
6.3 Results in Low-resource Scenarios

Given the complexity of argument structure analysis tasks and the difficulty in human annotation, data scarcity is a significant issue (Morio et al. 2022). For this reason, we carry out experiments in low-resource scenarios to show the superiority of UniASA with limited training data. Overall, aside from the F1_{ape} score for the APE task, UniASA outperforms all the current SOTA methods in the E2E-AM, APE, and AQE tasks.

Table 8
In-context learning results of UniASA (*w/o mv*) with GPT-4o (gpt-4o-2024-05-13).

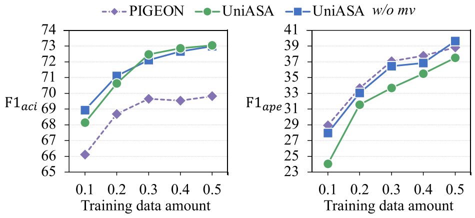
Task	Dataset	Setting	F1$_{aci}$ Micro	F1$_{ari}$ Micro	F1$_{arict}$	F1$_{ape}$	F1$_{aqe}$ Test
E2E-AM	AAEC	2-shot	55.42	30.00	24.20	–	–
		4-shot	57.27	31.66	26.77	–	–
		8-shot	59.90	33.32	28.51	–	–
		16-shot	**61.37**	**33.59**	30.58	–	–
		32-shot	61.06	32.36	**30.87**	–	–
	CDCP	2-shot	48.86	10.77	7.63	–	–
		4-shot	51.13	14.86	9.48	–	–
		8-shot	53.76	18.79	12.85	–	–
		16-shot	57.18	**21.64**	**16.81**	–	–
		32-shot	**57.53**	20.54	15.22	–	–
	AbstRCT	2-shot	56.77	32.22	30.78	–	–
		4-shot	57.08	31.78	30.85	–	–
		8-shot	61.43	37.70	35.50	–	–
		16-shot	63.84	**38.63**	**36.15**	–	–
		32-shot	**63.89**	38.24	35.67	–	–
	MTC	2-shot	68.41	38.17	34.03	–	–
		4-shot	70.34	42.05	38.24	–	–
		8-shot	70.61	44.95	41.54	–	–
		16-shot	75.21	**52.14**	**49.40**	–	–
		32-shot	**76.64**	51.43	48.78	–	–
	AASD	2-shot	57.92	47.48	31.35	–	–
		4-shot	59.56	48.22	32.63	–	–
		8-shot	65.12	53.00	39.36	–	–
		16-shot	67.55	**56.55**	44.67	–	–
		32-shot	**68.10**	55.51	**45.34**	–	–
APE	RR	2-shot	27.14	–	–	6.51	–
		4-shot	28.39	–	–	8.36	–
		8-shot	35.96	–	–	14.46	–
		16-shot	36.63	–	–	15.86	–
		32-shot	**37.21**	–	–	**16.46**	–
AQE	QAM	2-shot	–	–	–	–	6.95
		4-shot	–	–	–	–	10.65
		8-shot	–	–	–	–	11.44
		16-shot	–	–	–	–	11.24
		32-shot	–	–	–	–	**12.50**

The E2E-AM Task. Since the AAEC dataset is the most frequently used and representative dataset for the E2E-AM task, we test UniASA's performance in low-resource scenarios based on this dataset in Figure 6. It is evident that in low-resource scenarios, UniASA outperforms the current SOTA method, ST, in metrics F1$_{span}$, Micro F1$_{aci}$,



(a). *(b).* *(c).* *(d).*

Figure 6
Results in low-resource scenarios for the E2E-AM task on the AAEC dataset. The term "Training data amount" indicates the percentage of data randomly selected from the original training set for model training. Figures 6-*(a)* to *(d)* illustrate the four main metrics for the E2E-AM task: F1$_{span}$, Micro F1$_{aci}$, Micro F1$_{ari}$, and F1$_{arict}$. Here, Micro F1$_{aci}$ and Micro F1$_{ari}$ are reported instead of Macro F1$_{aci}$ and Macro F1$_{ari}$, as the former are more commonly used in the literature pertaining to the E2E-AM task (Eger, Daxenberger, and Gurevych 2017; Stab and Gurevych 2014).



(a). *(b).*

Figure 7
Results in low-resource scenarios for the APE task on the RR dataset.

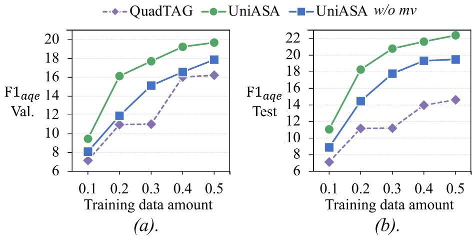
Micro F1$_{ari}$, and F1$_{arict}$, with multi-view learning playing a key role in this superior performance. Additionally, UniASA shows greater improvement in ACI-related metrics (F1$_{span}$ and Micro F1$_{aci}$). This suggests that using generative methods for the extraction and classification of argument components is a straightforward and effective strategy.

The APE Task. Figure 7 displays UniASA's performance in low-resource scenarios on the RR dataset. In terms of F1$_{aci}$, our method greatly surpasses the current SOTA model, PIGEON. Surprisingly, in terms of F1$_{ape}$, even though UniASA is similar to PIGEON in a full-data scenario (Table 5), it is notably inferior to PIGEON in low-resource scenarios (Figure 7-*(b)*). It is noteworthy that PIGEON elicits semantic knowledge from pre-trained models in advance to construct a probing graph, a method akin to incorporating external knowledge. We conjecture that this design may be more beneficial in low-resource scenarios. However, despite not incorporating such intricate designs, "UniASA *w/o mv*" still achieves results comparable to PIGEON. On the other hand, multi-view learning negatively impacts F1$_{ape}$, which aligns with our observation in Section 5.2.

The AQE Task. Figure 8 illustrates the results in low-resource scenarios on the QAM dataset. On both the validation and test sets, UniASA achieves significantly better results than the current SOTA method, QuadTAG. Importantly, we find that UniASA, with just 50% of the training data (Figure 8-*(b)*), already exceeds QuadTAG with full training data (Table 6) in terms of F1$_{aqe}$ on the test set. Such a finding highlights the robustness and superiority of UniASA.

6.4 Intermediate Fine-tuning

Similar to previous work (Morio et al. 2022; Liu et al. 2019a), we explore the impact of intermediate fine-tuning on improving the performance of UniASA. As different tasks



(a). *(b).*

Figure 8
Results in low-resource scenarios for the AQE task on the QAM dataset. "Val." is short for "Validation".

have distinct definitions and label sets, we first aggregate data from all three tasks for intermediate fine-tuning, followed by downstream fine-tuning on specific tasks. Notably, due to the extremely limited data samples in MTC and AASD, we exclude them from the intermediate fine-tuning stage, and use them only during downstream fine-tuning. As a result, the data used for intermediate fine-tuning comes from the five datasets: AAEC, CDCP, AbstRCT, RR, and QAM. Specifically, we first transform the data samples from the five datasets into source-target sequence pairs using the unified task template and the multi-view learning strategy. Then, all the transformed data samples are mixed together as intermediate fine-tuning data.[13] Note that because of the large amount of data in the RR dataset, which is over five times the size of the QAM dataset, we only sample 20% of data samples from the RR dataset for intermediate fine-tuning, to prevent the APE task from dominating this process. After intermediate fine-tuning, we further fine-tune the model on each specific dataset with the same method as in the main experiment. Furthermore, we also investigate how the choice of different intermediate fine-tuning tasks affects downstream tasks. Table 9 displays the results.

***Intermediate Fine-tuning on All Tasks*.** In general, intermediate fine-tuning on all tasks (*w. ift-all*) leads to a moderate improvement in UniASA's performance. For the E2E-AM task, "UniASA *w. ift-all*" achieves notable improvements on the AAEC, AbstRCT, MTC, and AASD datasets, but it does not have a significant impact on the CDCP dataset. We believe this is due to certain unique characteristics of the CDCP dataset, such as its domain (rulemaking) and relation density, which significantly differ from other datasets. For the APE task, since "UniASA *w/o mv*" performs better than UniASA in the main experiment (Table 5), we base the intermediate fine-tuning experiments of the APE task on "UniASA *w/o mv*". We can see that there is a noticeable improvement on the APE task. This suggests that combining data from the E2E-AM and AQE tasks can facilitate the APE task to some extent, despite differences in task definitions and data domains.

***The Choice of Different Intermediate Fine-tuning Tasks*.** We found that intermediate fine-tuning with a subset of tasks does not necessarily lead to inferior results compared to fine-tuning with all tasks. For example, on the QAM dataset, intermediate fine-tuning solely with data from the E2E-AM task yields the best results. On the AbstRCT dataset, intermediate fine-tuning with data from the APE and AQE tasks is the optimal choice.

6.5 Quantitative Error Analysis

We perform a detailed quantitative error analysis of UniASA's prediction results in the main experiment (Section 5). This analysis covers both the argument component identification subtask and the argumentative relation identification subtask. We analyze the results from two perspectives: *error rate* and *miss rate*. Error rate represents the proportion of incorrect predictions in the model's output, while miss rate refers to the proportion of gold labels that are not correctly predicted. Furthermore, errors are categorized into several types. Specifically, for the argument component identification subtask, a component position error refers to an incorrect prediction of an argument component's span or sentence position. A component type error, on the other hand,

[13] Here, special tokens are prepended to the source sequences to distinguish the data samples from different datasets, including "[AAEC]", "[CDCP]", "[AbstRCT]", "[RR]", "[QAM]".

Table 9
Results of intermediate fine-tuning. "*w. ift-all*" denotes the results of intermediate fine-tuning using data from all three tasks. "*w. ift-i*" denotes using data from task *i* for intermediate fine-tuning. For example, "*w. ift-e2eam+aqe*" indicates using data from both the E2E-AM and AQE tasks for intermediate fine-tuning.

Task	Dataset	Method	F1_{aci}	F1_{ari}	F1_{arict}	F1_{ape}	F1_{aqe}
			Micro	Micro			Test
E2E-AM	AAEC	UniASA	77.60	56.33	49.83	–	–
		w. ift-all	**78.24**	57.03	**50.87**	–	–
		w. ift-ape+aqe	77.88	**57.07**	50.34	–	–
		w. ift-ape	77.06	56.34	49.54	–	–
		w. ift-aqe	77.75	56.76	49.79	–	–
	CDCP	UniASA	68.16	**30.57**	**23.29**	–	–
		w. ift-all	**68.74**	30.47	22.66	–	–
		w. ift-ape+aqe	67.28	30.53	23.27	–	–
		w. ift-ape	67.25	29.93	20.39	–	–
		w. ift-aqe	67.62	28.42	21.47	–	–
	AbstRCT	UniASA	69.67	39.11	38.65	–	–
		w. ift-all	71.93	39.75	39.54	–	–
		w. ift-ape+aqe	72.54	**41.45**	**40.38**	–	–
		w. ift-ape	72.76	39.57	38.38	–	–
		w. ift-aqe	**74.22**	40.61	38.97	–	–
	MTC	UniASA	79.88	46.76	44.44	–	–
		w. ift-all	81.79	**47.93**	**46.43**	–	–
		w. ift-ape+aqe	82.10	45.22	43.07	–	–
		w. ift-ape	81.20	45.87	44.95	–	–
		w. ift-aqe	**82.43**	46.31	45.79	–	–
	AASD	UniASA	68.84	49.54	**44.78**	–	–
		w. ift-all	70.53	**51.88**	44.05	–	–
		w. ift-ape+aqe	68.13	48.06	42.30	–	–
		w. ift-ape	67.66	47.06	40.06	–	–
		w. ift-aqe	**70.88**	49.14	41.45	–	–
APE	RR	UniASA *w/o mv*	73.77	–	–	43.53	–
		w. ift-all	**74.30**	–	–	**44.30**	–
		w. ift-e2eam+aqe	73.45	–	–	43.86	–
		w. ift-e2eam	73.48	–	–	42.52	–
		w. ift-aqe	73.42	–	–	42.11	–
AQE	QAM	UniASA	–	–	–	–	23.52
		w. ift-all	–	–	–	–	24.38
		w. ift-e2eam+ape	–	–	–	–	24.62
		w. ift-e2eam	–	–	–	–	**24.68**
		w. ift-ape	–	–	–	–	24.34

occurs when the component's position is correctly predicted, but its type is misclassified. For the argumentative relation identification subtask, if an argumentative relation is marked as having a component position error, it means that at least one of its two related components has a position error. Similarly, if an argumentative relation is marked as having a component type error, it means that both of its two components' positions are correctly predicted, but at least one of their types is misclassified. Finally, a relation type error indicates that while the positions of both components in the argumentative relation are correctly predicted, the type of this relation is incorrectly predicted. Notably, component type errors and relation type errors in the argumentative

Table 10
Quantitative error analysis results. The "Baseline" column refers to the ST model for the E2E-AM task, the PIGEON model for the APE task, and the QuadTAG model for the AQE task, respectively. The predictions of all baseline models are reproduced using their official code.

Task	**Dataset**	**Metric**	Component Identification			Relation Identification		
			Baseline	UniASA	Δ	Baseline	UniASA	Δ
E2E-AM	AAEC	Error Rate (%)	26.03	21.17	−4.86	53.01	45.94	−7.07
		Component Position Error	16.66	11.93	−4.73	20.45	15.66	−4.79
		Component Type Error	9.37	9.24	−0.13	20.57	16.51	−4.06
		Relation Type Error	–	–	–	23.86	23.77	−0.09
		Miss Rate (%)	22.62	23.57	+0.95	53.82	53.51	−0.31
	CDCP	Error Rate (%)	33.04	33.04	0.00	69.75	68.00	−1.75
		Component Position Error	19.27	19.41	+0.14	27.36	27.55	+0.19
		Component Type Error	13.77	13.63	−0.14	16.05	19.51	+3.46
		Relation Type Error	–	–	–	34.59	29.40	−5.19
		Miss Rate (%)	30.38	32.63	+2.25	78.53	84.01	+5.48
	AbstRCT	Error Rate (%)	39.74	28.76	−10.98	67.42	60.91	−6.51
		Component Position Error	34.11	23.34	−10.77	41.37	28.24	−13.13
		Component Type Error	5.63	5.42	−0.21	5.63	5.28	−0.35
		Relation Type Error	–	–	–	23.67	30.85	+7.18
		Miss Rate (%)	30.66	28.80	−1.86	63.28	63.94	+0.66
	MTC	Error Rate (%)	19.71	16.74	−2.97	56.80	45.36	−11.44
		Component Position Error	13.08	9.06	−4.02	20.75	12.36	−8.39
		Component Type Error	6.63	7.68	+1.05	9.66	9.83	+0.17
		Relation Type Error	–	–	–	34.98	31.37	−3.61
		Miss Rate (%)	20.16	23.21	+3.05	57.07	62.51	+5.44
	AASD	Error Rate (%)	26.93	25.35	−1.58	52.19	45.92	−6.27
		Component Position Error	11.31	9.09	−2.22	17.17	12.17	−5.00
		Component Type Error	15.62	16.26	+0.64	19.96	18.42	−1.54
		Relation Type Error	–	–	–	28.73	25.27	−3.46
		Miss Rate (%)	28.71	32.21	+3.50	53.46	60.78	+7.32
APE	RR	Error Rate (%)	28.06	25.18	−2.88	58.94	56.24	−2.70
		Component Position Error	28.06	25.18	−2.88	44.11	41.35	−2.76
		Relation Type Error	–	–	–	14.83	14.89	+0.06
		Miss Rate (%)	26.83	26.66	−0.17	55.16	57.67	+2.51
AQE	QAM	Error Rate (%)	45.95	41.03	−4.92	75.63	69.32	−6.31
		Component Position Error	30.43	29.78	−0.65	36.44	36.41	−0.03
		Component Type Error	15.53	11.25	−4.28	30.38	25.10	−5.28
		Relation Type Error	–	–	–	30.34	29.23	−1.11
		Miss Rate (%)	56.55	58.45	+1.90	82.15	80.71	−1.44

relation identification subtask may overlap. We believe that this categorization method can effectively reflect the model's predictive capabilities for the two subtasks.

The results of the quantitative error analysis are presented in Table 10. Compared to the baseline methods, UniASA achieves a significant reduction in error rate across the majority of datasets. This demonstrates that UniASA can make more accurate and reliable predictions. For the argument component identification subtask, particularly on the E2E-AM task, the primary reduction in errors comes from the component position errors, while the reduction in the component type errors is not significant in most cases. This may be because classifying component types is a simpler task compared to identifying component positions, so the baselines already perform relatively well in this area. Moreover, regarding the argumentative relation identification subtask, UniASA's performance in reducing the relation type errors does not show significant improvement on most datasets. This indicates that the identification of relations remains a considerable challenge. The reason may be the relatively small amount of training data related to argumentative relations (as shown in Table K.1), which could potentially be improved through data augmentation specifically targeting relations. For the miss rate, UniASA performs worse compared to the baselines, especially on the E2E-AM and APE tasks. This suggests that the baselines, ST and PIGEON, tend to make more predictions, while UniASA is more conservative in its predictions. This may be related to the voting mechanism in our multi-view learning strategy. Overall, we conclude that UniASA's predictions are more accurate compared to the baselines, but the downside is a higher miss rate. This may suggest that the model only makes predictions when it is highly confident, resulting in fewer predictions but with higher accuracy.

6.6 Case Study and Qualitative Error Analysis

We conduct case studies and qualitative error analysis to further illustrate the strengths and limitations of UniASA. The results demonstrate that while UniASA's performance is not perfect, it generally outperforms the baseline models. For more details, please refer to Appendix H.

7. Conclusion

We present UniASA, a unified generative framework for argument structure analysis. This framework is capable of uniformly addressing three tasks: end-to-end argument mining, argument pair extraction, and argument quadruplet extraction. Furthermore, we propose a subtask decomposition-based multi-view learning strategy to enhance UniASA. Extensive experiments on seven datasets demonstrate the superiority of UniASA. Further analysis shows that UniASA performs well in low-resource scenarios. We also show that by fine-tuning or in-context learning, UniASA can be effectively integrated with large language models such as Llama, and fine-tuning, in particular, leads to significantly improved performance.

8. Limitations

Although our proposed UniASA can effectively unify various argument structure analysis tasks and achieve good performance, it still has some limitations.

First, as discussed in Section 5.1 and Appendix F, for the E2E-AM task, UniASA is less effective at learning relation categories with sparse data compared with the baseline model ST, leading to a slightly lower Macro $F1_{ari}$ score. Second, as mentioned in Section 6.5, UniASA shows limited improvement in reducing relation type errors across most datasets, and in some cases, it even underperforms the baselines. Both of these issues can be attributed to data imbalance and data scarcity, and could potentially be alleviated through targeted data augmentation techniques. Additionally, compared with the baselines, UniASA has a higher miss rate. Given its lower error rate, we believe that UniASA tends to make predictions with higher confidence, resulting in fewer but more accurate predictions overall. We speculate that adopting different decoding strategies or adjusting decoding parameters to generate longer outputs might help alleviate this issue. In future work, we plan to explore solutions to address these limitations.

Appendix A. Results for the E2E-AM Task with Oracle Spans

We show the main experimental results for the E2E-AM task with oracle argument component spans in Table A.1. As can be seen, for all five datasets, UniASA surpasses ST on the Micro F1$_{aci}$, Macro F1$_{ari}$, and F1$_{arict}$ metrics. However, UniASA does not perform well on macro metrics, particularly on Macro F1$_{ari}$. This phenomenon is consistent with our finding in Section 5.1, that is, our method's ability to process data-scarce categories is slightly inferior to ST's. Also, although removing multi-view learning from UniASA (*w/o mv*) leads to a decline in overall performance, it still outperforms ST in F1$_{arict}$ on the AAEC and AbstRCT datasets.

Table A.1
Main results for the E2E-AM task with oracle argument component spans. * denotes statistical significance with $p < 0.05$. $^\diamond$ indicates the current SOTA method. † denotes that the evaluation results are computed by the metric from the studies of Kuribayashi et al. (2019) and Bao et al. (2021a).[14]

Dataset	Method	F1$_{aci}$		F1$_{link}$	F1$_{ari}$		F1$_{arict}$
		Micro	Macro		Micro	Macro	
AAEC	ST$^\diamond$	87.44	79.55	67.16	66.29	**55.95**	55.44
	UniASA	**89.33***	**82.54***	**69.42***	**67.71***	54.35	**58.43***
	w/o mv	88.47	81.70	68.11	65.81	51.28	57.27
AAEC (*para.*)	Joint-ILP†	–	82.60	58.50	–	–	–
	Ptr. Net.†	–	84.90	60.80	–	–	–
	Span-LSTM†	–	85.70	67.80	–	–	–
	BERT-Trans†	–	88.40	**70.60**	–	–	–
	ST$^{\dagger\diamond}$	88.40	86.82	69.33	68.14	**57.11**	57.60
	UniASA†	**89.15***	88.60*	69.80	**69.26***	56.24	**58.77***
	w/o mv†	88.63	**89.12**	69.49	67.68	54.13	58.50
CDCP	TSP-PLBA	–	78.91	–	34.04	–	–
	BERT-Trans	–	82.50	–	37.30	–	–
	ST$^\diamond$	81.03	82.34	40.15	40.11	20.39	28.70
	UniASA	**82.40***	**84.03***	**40.28**	**40.28**	**20.50**	**28.75**
	w/o mv	82.20	82.73	38.29	38.25	19.46	28.47
AbstRCT	Rel.RoBERTa	–	–	–	48.72	17.53	–
	Rel.SciBERT	–	–	–	58.21	36.76	–
	ST$^\diamond$	89.37	67.57	59.65	57.10	47.12	52.35
	UniASA	**89.90**	**67.79**	**60.77***	**58.50***	**48.01***	**54.52***
	w/o mv	89.18	65.73	60.14	57.49	47.25	53.46
MTC	ST$^\diamond$	89.85	**83.70**	**65.06**	55.23	39.27	52.72
	UniASA	**90.21**	83.26	63.97	**56.13**	**41.73***	**53.38**
	w/o mv	89.38	82.12	63.13	53.85	38.53	52.85
AASD	ST$^\diamond$	77.78	65.90	63.96	58.30	49.22	50.29
	UniASA	**81.59***	**67.73***	73.00*	**61.92***	**52.46***	**50.55**
	w/o mv	78.51	63.47	**73.16***	59.98	51.72*	48.64

[14] This evaluation metric maps both the "Claim:Against" and "Claim:For" types in the AAEC dataset to "Claim". Apart from this, everything else remains the same as our evaluation approach introduced in Section 4.3.

Appendix B. Implementation Details for Fine-tuning Larger Language Models

For the t5-large model, we use an approach similar to that for fine-tuning the t5-base (Section 4.2), with only slight adjustments to the hyperparameters, as shown in Table B.1.

For the t5-3b and Llama-3-8B models, we use the QLoRA technique (Dettmers et al. 2023) for fine-tuning, implemented with the PEFT,[15] transformers,[16] and bitsandbytes[17] libraries. The specific QLoRA-related parameters are shown in Table B.2, while other regular parameters are listed in Table B.1.

Table B.1
Hyperparameters for the experiments of fine-tuning larger language models.

Model	Task	Dataset	Learning Rate	Batch Size	Epoch	Warm-up Ratio	Weight Decay
t5-large	E2E-AM	AAEC	1e-4	2	30	0.1	0.0
		CDCP	1e-4	2	30	0.1	0.0
		AbstRCT	1e-4	2	10	0.1	0.0
		MTC	2e-4	4	150	0.0	0.0
		AASD	1e-4	2	150	0.1	0.0
	APE	RR	1e-4	2	10	0.1	0.1
	AQE	QAM	1e-4	2	8	0.1	0.1
t5-3b	E2E-AM	AAEC	2e-4	2	20	0.1	0.0
		CDCP	2e-4	2	20	0.1	0.0
		AbstRCT	2e-4	2	20	0.1	0.0
		MTC	2e-4	2	60	0.0	0.0
		AASD	2e-4	2	50	0.1	0.0
	APE	RR	2e-4	2	5	0.0	0.0
	AQE	QAM	2e-4	2	8	0.1	0.1
llama-3-8b	E2E-AM	AAEC	2e-4	4	30	0.1	0.0
		CDCP	2e-4	4	30	0.1	0.0
		AbstRCT	2e-4	4	20	0.1	0.0
		MTC	2e-4	4	60	0.0	0.0
		AASD	2e-4	4	100	0.0	0.0
	APE	RR	2e-4	2	5	0.0	0.0
	AQE	QAM	2e-4	4	8	0.1	0.1

[15] https://github.com/huggingface/peft.

[16] https://github.com/huggingface/transformers.

[17] https://github.com/bitsandbytes-foundation/bitsandbytes.

Table B.2
QLoRA hyperparameters for the experiments of fine-tuning larger language models.

Hyperparameters	t5-3b	llama-3-8b
lora_alpha	32	32
lora_dropout	0.05	0.1
r	16	64
bias	"none"	"none"
task_type	"SEQ_2_SEQ_LM"	"CAUSAL_LM"
target_modules	"q v k o"	"q_proj k_proj v_proj o_proj"
load_in_4bit	True	True
bnb_4bit_quant_type	"nf4"	"nf4"
bnb_4bit_compute_dtype	torch.bfloat16	torch.bfloat16
bnb_4bit_use_double_quant	True	True
optimizer	paged_adamw_8bit	paged_adamw_8bit

Appendix C. Discussion on Potential Data Leakage of Using T5

Data leakage is an important issue in the NLP community. Because the backbone architecture of UniASA is the pre-trained T5 models, it is necessary to discuss the potential data leakage issues that may exist due to the use of T5. Therefore, we conduct a detailed analysis and conclude that *the T5 models have not been fine-tuned or pre-trained on the seven datasets used in our article*. Specifically, according to the description in the original T5 paper (Raffel et al. 2020), the training of the T5 models can be divided into two steps: (1) pre-training on the C4 dataset by an unsupervised denoising (language modeling) task, and (2) fine-tuning on downstream task datasets. Next, we discuss potential data leakage issues for these two steps separately.

Downstream Task Fine-tuning of the T5 Models. Section "2.3 Downstream Tasks" of the original T5 paper explicitly lists the downstream tasks and datasets used for fine-tuning, which include: text classification meta-benchmarks (GLUE and SuperGLUE), abstractive summarization (CNN/Daily Mail), question answering (SQuAD), and translation (WMT English to German, French, and Romanian). These tasks and datasets are unrelated to the argument structure analysis datasets used in our article. Thus, we can confidently assert that the T5 models were not fine-tuned on any of the seven datasets used in our paper during the downstream task fine-tuning process. Similarly, according to the original LongT5 (Guo et al. 2022) paper, the LongT5 models were also not fine-tuned on these seven datasets.

Pre-training of the T5 Models on the C4 Dataset. According to Section 3.1.4 (Unsupervised Objective) of the original T5 paper, the "denoising objective" for pre-training T5 utilizes unlabeled data. Therefore, during this process, it is impossible for T5 to have been directly trained on the annotated datasets used in our article. However, there remains a possibility that T5 has "seen" the annotated datasets used in our article through the denoising task during pre-training. To eliminate this possibility, we conducted the following analysis on the seven datasets used in our article.

We first confirm that the annotated AbstRCT (2020), AASD (2021), RR (2020), and QAM (2023) datasets were released after the original T5 paper (2019) (Raffel et al. 2020). Therefore, it is impossible that these datasets' annotation information was used during the training of the T5 models.

For the AAEC, MTC, and CDCP datasets, we conduct analysis by retrieving similar data from the T5 models' pre-training dataset C4. Specifically, we obtain the T5 models' pre-training data from `https://huggingface.co/datasets/allenai/c4`, which totals 305GB. We then build a retrieval system based on BM25 using the pyserini library,[18] indexing the C4 dataset and using the input text of each data entry from the AAEC, MTC, and CDCP datasets as query.[19] Through this retrieval system, we retrieve the top 3 most similar C4 data entries for each query from the AAEC, MTC, and CDCP datasets. We then calculate the "C4 Coverage" score, a metric that represents the proportion of 4-grams in the query that also appear in its retrieved C4 data. The higher the "C4 Coverage" score, the more text from the query appears in the retrieved C4 data entry. When the "C4 Coverage" score is 100%, it indicates that the query is completely present in its retrieved C4 data entry. In practice, a "C4 Coverage" score exceeding 10% often suggests that at least one sentence in the query appears in its retrieved C4 data entry. We find that the "C4 Coverage" score for each test data entry in the MTC and CDCP datasets is significantly below 10% (with the highest being 8.30% and 6.45%, respectively). Therefore, we believe that there is no risk of data leakage for the MTC and CDCP datasets. For the AAEC dataset, we discover that only 3 out of 80 test data entries have a "C4 Coverage" score exceeding 10% (76.94%, 30.59%, and 12.25%, respectively). We further manually inspect these three data entries and their retrieved C4 data entries, and find that despite the slightly higher "C4 Coverage", the retrieved C4 data entries do not directly contain any annotation information related to the AAEC dataset. Therefore, we can conclude that the test data in the AAEC, MTC, and CDCP datasets were not leaked during T5's pre-training.

In conclusion, the seven datasets used in this article were not leaked during T5's pre-training. Additionally, since LongT5 was also pre-trained using the C4 dataset, the same conclusion applies to LongT5 as well.

[18] `https://github.com/castorini/pyserini`.

[19] Note that for AAEC and CDCP, we only analyze their test sets, as "data leakage" generally refers to the leakage of test data. Since MTC does not have a designated test set, we analyze all of its data.

Appendix D. More In-context Learning Results

The in-context learning results of Llama-3-70B-Instruct and Mixtral-8×7B-Instruct-v0.1 are shown in Table D.1. Due to the maximum context window size of 8k for Llama-3-70B-Instruct, it only supports up to a 4-shot setting. Mixtral-8×7B-Instruct-v0.1, with a maximum context window size of 32k, can support a 16-shot setting except for the RR dataset. Overall, although Mixtral-8×7B-Instruct-v0.1 has a longer context window size than Llama-3-70B-Instruct, its performance is not as strong. We suspect that this may be attributed to the limited number of model parameters of Mixtral-8×7B-Instruct-v0.1.

Table D.1
In-context learning results of UniASA (*w/o mv*) with Llama-3-70B-Instruct and Mixtral-8×7B-Instruct-v0.1.

Task	Dataset	Model	Setting	F1_{aci} Micro	F1_{ari} Micro	F1_{arict}	F1_{ape}	F1_{aqe} Test
E2E-AM	AAEC	Llama-3-70B-Instruct	2-shot	50.01	23.56	15.64	–	–
			4-shot	**50.84**	**26.28**	**20.95**	–	–
		Mixtral-8×7B-Instruct-v0.1	2-shot	43.67	19.05	12.23	–	–
			4-shot	41.93	20.49	14.77	–	–
			8-shot	49.51	23.19	19.35	–	–
			16-shot	47.34	23.75	17.30	–	–
	CDCP	Llama-3-70B-Instruct	2-shot	42.93	11.05	6.35	–	–
			4-shot	**46.05**	**16.90**	8.16	–	–
		Mixtral-8×7B-Instruct-v0.1	2-shot	35.54	7.05	3.77	–	–
			4-shot	38.03	7.71	3.78	–	–
			8-shot	42.87	13.81	**9.52**	–	–
			16-shot	44.64	13.24	8.96	–	–
	AbstRCT	Llama-3-70B-Instruct	2-shot	48.71	23.79	12.85	–	–
			4-shot	**61.63**	**33.13**	**32.42**	–	–
		Mixtral-8×7B-Instruct-v0.1	2-shot	42.92	19.48	18.67	–	–
			4-shot	46.28	25.25	23.46	–	–
			8-shot	50.91	26.56	24.41	–	–
			16-shot	54.63	31.23	29.80	–	–
	MTC	Llama-3-70B-Instruct	2-shot	68.26	39.92	36.07	–	–
			4-shot	69.98	**42.00**	**39.82**	–	–
		Mixtral-8×7B-Instruct-v0.1	2-shot	61.61	28.99	27.58	–	–
			4-shot	62.81	33.76	30.42	–	–
			8-shot	68.88	36.37	34.77	–	–
			16-shot	**71.97**	40.10	38.62	–	–
	AASD	Llama-3-70B-Instruct	2-shot	56.42	48.40	26.72	–	–
			4-shot	62.41	**50.71**	32.05	–	–
		Mixtral-8×7B-Instruct-v0.1	2-shot	52.64	31.39	16.71	–	–
			4-shot	59.17	35.27	23.79	–	–
			8-shot	60.16	39.43	29.65	–	–
			16-shot	**66.17**	49.08	**40.19**	–	–
APE	RR	Llama-3-70B-Instruct	2-shot	6.87	–	–	1.14	–
			4-shot	9.17	–	–	1.84	–
		Mixtral-8×7B-Instruct-v0.1	2-shot	17.00	–	–	1.32	–
			4-shot	18.33	–	–	2.41	–
			8-shot	**18.36**	–	–	**4.15**	–
AQE	QAM	Llama-3-70B-Instruct	2-shot	–	–	–	–	7.32
			4-shot	–	–	–	–	**8.55**
		Mixtral-8×7B-Instruct-v0.1	2-shot	–	–	–	–	5.63
			4-shot	–	–	–	–	5.99
			8-shot	–	–	–	–	6.45
			16-shot	–	–	–	–	5.76

Appendix E. Implementation Details and Prompt Templates for In-context Learning

E.1 Implementation Details

All prompt templates used for in-context learning are shown in Section E.2. We also present a complete example of a 2-shot prompt for the AAEC dataset in Section E.3. For each experiment, the few-shot demonstrations are randomly selected from the training set. Note that for AASD and MTC, since they do not have an official training/test split, we randomly select demonstrations from all data samples and test on all other data samples except those chosen as demonstrations. For the other datasets, we report results on their test sets. Each experiment is conducted with 3 different random seeds and different demonstrations, and the average results are reported.

For all models, the temperature is set to 0.0 to ensure the output is as deterministic as possible.[20] The system prompt is "You are a powerful argument structure analyzer who can clearly analyze the argument structure of the input argumentative documents." The specific API version of the GPT-4o model is gpt-4o-2024-05-13.[21] Llama-3-70B-Instruct[22] and Mixtral-8×7B-Instruct-v0.1[23] are obtained from `huggingface.co/models`. We use the vLLM[24] library to implement model inference with 2 NVIDIA H100 GPUs. The decoding strategy used is the default sampling decoding in the vLLM library and the OpenAI API.

E.2 Prompt Template

In the in-context learning experiments, for all models and tasks, the prompt template we used is shown as follows. We design prompts with the principle of keeping them basic and concise, in order to reflect a baseline result of in-context learning.

Here, **{component_category_set}** and **{relation_category_set}** refer to the specific component category set and relation category set for each dataset, as shown in Table 1. **{example_i_source_sequence}** and **{example_i_target_sequence}** represent the source sequence and target sequence of the randomly selected demonstration sample, respectively. Specific examples of the source sequence and target sequence can be found in Figure 4. **{source_sequence}** refers to the source sequence of the current sample being processed. A complete example of a 2-shot prompt for the AAEC dataset is shown in Section E.3.

[20] `https://platform.openai.com/docs/advanced-usage/reproducible-outputs`.
[21] `https://platform.openai.com/docs/models/gpt-4o`.
[22] `https://huggingface.co/meta-llama/Meta-Llama-3-70B-Instruct`.
[23] `https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1`.
[24] `https://github.com/vllm-project/vllm`.

Please do an argument structure analysis task. Given an input document, output its argument structure. Note that the types of argument components contain **{component_category_set}**, while the types of argumentative relations contain **{relation_category_set}**.

Examples are shown below:

Example 1:

Input:
{example_1_source_sequence}

Output:
{example_1_target_sequence}

Example 2:

Input:
{example_2_source_sequence}

Output:
{example_2_target_sequence}

...

Based on the above example, process the following text:
(Please strictly follow the format of the output in the examples. Do not include any other information in the output.)

Input:
{source_sequence}

Output:

E.3 An Example of Prompt

Here, we present a complete example of a 2-shot prompt for the AAEC dataset. Note that [NEW_LINE] is a special token we defined to represent a line break.

Please do an argument structure analysis task. Given an input document, output its argument structure. Note that the types of argument components contain {'MajorClaim', 'Claim:Against', 'Premise', 'Claim:For'}, while the types of argumentative relations contain {'Support', 'Attack'}.

Examples are shown below:

Example 1:

Input:
[ASA] Please do an argumentation structure analysis task. | Topic: Dancing plays an important role in a culture. | Text: In every country, people do dance for fun, for entertrain, and for decrease a stress. Dancing is also one of important parts of culture that represent to something that people believe. Many cultures do dance to show a great prosperous of its culture. And also dancing to entertain people. I will give reasons to support my opinion. [NEW_LINE] First, dancing are significant part of culture that could show to something that people believe. For example, some culture people believe that they dance in order to sacrifice to their god so they will be get a happy life. Or some part of the world that are in arid area believe that if they danced, the rain would come. Consequently, dancing are the part of people belief. [NEW_LINE] Secondly, dancing can represent to civilization of that culture. For instance, I have seen many western movies that when a kingdom have visitor from another kingdom, king or queen would dance to show a splendid of their kingdom culture. The way of movement, the way that they costume and also a music while they dance. All this is a civilization of their culture. [NEW_LINE] Thirdly, many culture, people have many ways to entertain themselves. And dancing is one of that ways. Both people who see and dance also have a good time. For example, many event, such as christmas, last christmas I went to mall to buy gifts, and I was entertained by group of people in santa custom dancing in christmas song. I felt that people around me happy and group of santas also happy to see people happy. [NEW_LINE] In conclusion, I strongly agree with notion that dancing is an important part of culture because dancing are represent to believe of people. Besides, dancing could show how civilization of that culture is. Moreover, people use dancing to entertain themselves. So dancing is important in every culture.

Output:
[ASA] The type of [# dancing are significant part of culture that could show to something that people believe #] is "Premise". It is a "Support" type of argument for [# dancing are the part of people belief #] ("Claim:For"). [SEP] The type of [# some culture people believe that they dance in order to sacrifice to their god so they will be get a happy life #] is "Premise". It is a "Support" type of argument for [# dancing are the part of people belief #] ("Claim:For"). [SEP] The type of [# Or some part of the world that are in arid area believe that if they danced, the rain would come #] is "Premise". It is a "Support" type of argument for [# dancing are the part of people belief #] ("Claim:For"). [SEP] The type of [# dancing are the part of people belief #] is "Claim:For". It produces no relation. [SEP] The type of [# dancing can represent to civilization of that culture #] is "Claim:For". It produces no relation. [SEP] The type of [# I have seen many western movies that when a kingdom have visitor from another kingdom, king or queen would dance to show a splendid of their kingdom culture #] is "Premise". It is a "Support" type of argument for [# dancing can represent to civilization of that culture #] ("Claim:For"). [SEP] The type of [# The way of movement, the way that they costume and also a music while they dance #] is "Premise". It is a "Support" type of argument for [# dancing can represent to civilization of that culture #] ("Claim:For"). [SEP] The type of [# All this is a civilization of their culture #] is "Premise". It is a "Support" type of argument for [# dancing can represent to civilization of that culture #] ("Claim:For"). [SEP] The type of [# many culture, people have many ways to entertain themselves #] is "Premise". It is a "Support" type of argument for [# dancing is one of that ways #] ("Claim:For"). [SEP] The type of [# dancing is one of that ways #] is "Claim:For". It produces no relation. [SEP] The type of [# Both people who see and dance also have a good time #] is "Premise". It is a "Support" type of argument for [# dancing is one of that ways #] ("Claim:For"). [SEP] The type of [# many event, such as christmas, last christmas I went to mall to buy gifts, and I was entertained by group of people in santa custom dancing in christmas song #] is "Premise". It is a "Support" type of argument for [# dancing is one of that ways #] ("Claim:For"). [SEP] The type of [# I felt that people around me happy and group of santas also happy to see people happy #] is "Premise". It is a "Support" type of argument for [# dancing is one of that ways #] ("Claim:For"). [SEP] The type of [# dancing is an important part of culture #] is "MajorClaim". It produces no relation. [SEP] The type of [# dancing are represent to believe of people #] is "Claim:For". It produces no relation. [SEP] The type of [# dancing could show how civilization of that culture is #] is "Premise". It is a "Support" type of argument for [# dancing is important in every culture #] ("Claim:For"). [SEP] The type of [# people use dancing to entertain themselves #] is "Premise". It is a "Support" type of argument for [# dancing is important in every culture #] ("Claim:For"). [SEP] The type of [# dancing is important in every culture #] is "Claim:For". It produces no relation. [SEP]

Example 2:

Input:
[ASA] Please do an argumentation structure analysis task. | Topic: Which is better- Learning foreign language in home or host country? | Text: Our World has varied cultures and each culture has its own language. It has become essential for a person to learn at least one foreign language as one do interacts with foreigners once in a lifetime and learning a foreign langauge adds on to a person skill-set as well. Also, I think its better that people learn foreign language in their own country. The reasons for holding that opinion are illustrated in the subsequent paragraphs. [NEW_LINE] First of all, it is not possible for everyone to visit a foreign country to learn a foreign language as it requires large amount of money and legal formalities. In the earlier periods, it was not easy to find a teacher to learn a foreign languge. But with the advent of globalization, one can easily find a teacher in their own country. For example, in Indian sub-continent there are large number of centers opened for getting trained in variety of languages like french, german etc. [NEW_LINE] Second, colleges and universities have introduced foreign language course as a part of their curriculum. Students can choose any language from the set of languages available at a university. So, it is very easy to learn a foreign in our own country. Also, highly qualified foreign language teachers are easily accessible in various schools and colleges. [NEW_LINE] Third, technology has also played a great role in reducing the separation between learning a foreign languge in our own country. A large number of foreign language courses are available online and there are various forums where people can discuss their problems on different foreign languages. [NEW_LINE] To sum up, it is better for people to learn a foreign language in their own country. It is cost effective and there is a pool of opportunities available now to accomplish the same.

Output:
[ASA] The type of [# its better that people learn foreign language in their own country #] is "MajorClaim". It produces no relation. [SEP] The type of [# it is not possible for everyone to visit a foreign country to learn a foreign language #] is "Claim:For". It produces no relation. [SEP] The type of [# it requires large amount of money and legal formalities #] is "Premise". It is a "Support" type of argument for [# it is not possible for everyone to visit a foreign country to learn a foreign language #] ("Claim:For"). [SEP] The type of [# In the earlier periods, it was not easy to find a teacher to learn a foreign languge #] is "Premise". It is an "Attack" type of argument for [# with the advent of globalization, one can easily find a teacher in their own country #] ("Claim:For"). [SEP] The type of [# with the advent of globalization, one can easily find a teacher in their own country #] is "Claim:For". It produces no relation. [SEP] The type of [# in Indian sub-continent there are large number of centers opened for getting trained in variety of languages like french, german etc #] is "Premise". It is a "Support" type of argument for [# with the advent of globalization, one can easily find a teacher in their own country #] ("Claim:For"). [SEP] The type of [# colleges and universities have introduced foreign language course as a part of their curriculum #] is "Premise". It is a "Support" type of argument for [# it is very easy to learn a foreign in our own country #] ("Claim:For"). [SEP] The type of [# Students can choose any language from the set of languages available at a university #] is "Premise". It is a "Support" type of argument for [# it is very easy to learn a foreign in our own country #] ("Claim:For"). [SEP] The type of [# it is very easy to learn a foreign in our own country #] is "Claim:For". It produces no relation. [SEP] The type of [# highly qualified foreign language teachers are easily accessible in various schools and colleges #] is "Premise". It is a "Support" type of argument for [# it is very easy to learn a foreign in our own country #] ("Claim:For"). [SEP] The type of [# technology has also played a great role in reducing the separation between learning a foreign languge in our own country #] is "Claim:For". It produces no relation. [SEP] The type of [# A large number of foreign language courses are available online and there are various forums where people can discuss their problems on different foreign languages #] is "Premise". It is a "Support" type of argument for [# technology has also played a great role in reducing the separation between learning a foreign languge in our own country #] ("Claim:For"). [SEP] The type of [# it is better for people to learn a foreign language in their own country #] is "MajorClaim". It produces no relation. [SEP] The type of [# It is cost effective and there is a pool of opportunities available now to accomplish the same #] is "Claim:For". It produces no relation. [SEP]

Based on the above example, process the following text:
(Please strictly follow the format of the output in the examples. Do not include any other information in the output.)

Input:
[ASA] Please do an argumentation structure analysis task. | Topic: Study at school or get a job? | Text: Many people believe that children should study at school to have more knowledge that prepare better for their future. Others, however, think that these children may disrupt their school work and should be allowed to leave school early to find a job. Personally, I tend to agree with the point of view that student have to be forced to study at school. [NEW_LINE] First of all, schools offer to students a good environment with experienced professors and high quality programs for studying. It creates the best conditions for students education and can force them to focus on their school work instead of wasting their time to do useless things. Second of all, schools provide lots of academic knowledge to students. Students may learn professional skills, expand their understandings and gain experiences. Therefore, they have more opprotunities to find a job and to be successful in the future. For example, as we know, employer always prefer to hire an employee of high degree who have professional skills. [NEW_LINE] Nevertheless, it is not unreasonable that some people think that children should interrupt their school work and get a job. Whether children can learn a lot at school, there are many subjects that will be of little value to them in the future. Furthermore, children can learn social skills when they have a job. They can get more experiences that can not be obtained at school. Working helps children be more independent and teach them to esteem and manage the money that they've earned. [NEW_LINE] Overall, I believe that students should study at school. Even though there are some advantages of leaving school to find a job, studying at school is always the best choice for children's future. There are many ways that can train children to learn independent and social skills instead of getting a job.

Output:

Appendix F. Detailed Main Experimental Results for the E2E-AM Task

We present the detailed main experimental results for the E2E-AM task in Tables F.1–F.5, specifically showing the F1 scores for each component and relation category. We also present the number of instances (# Instances) of each component and relation category that appear in the dataset, where the "Macro" column indicates the total number of components or relations. Notably, since the original authors of ST did not report F1 scores for each individual category in their paper, the ST results shown here are reproduced using their official code.

Similar to our findings in Section 5, UniASA shows weaker performance compared with the ST model in learning data-scarce categories, particularly for the "Attack" type in the AAEC dataset, and the "Value" and "Partial-Attack" types in the AbstRCT dataset. However, for other categories with more abundant data, UniASA can outperform ST. Therefore, overall, UniASA can achieve performance that is either better than or on par with ST.

Table F.1
Detailed main experimental results on the AAEC dataset.

Dataset	Method	$F1_{aci}$					$F1_{ari}$		
		MajorClaim	Claim:For	Claim:Against	Premise	Macro	Support	Attack	Macro
AAEC	ST	76.06	64.15	**44.45**	82.12	66.70	54.65	**30.36**	**42.51**
	UniASA	**76.69**	**67.61**	43.80	**83.14**	**67.81**	**58.13**	25.11	41.62
# Instances		751	1,228	278	3,832	6,089	3,613	219	3,832

Table F.2
Detailed main experimental results on the CDCP dataset.

Dataset	Method	$F1_{aci}$						$F1_{ari}$		
		Value	Fact	Policy	Testimony	Reference	Macro	Reason	Evidence	Macro
CDCP	ST	**72.15**	**48.38**	79.31	**63.52**	65.67	65.81	**32.00**	0.00	**16.00**
	UniASA	69.30	45.78	**83.33**	63.16	**100.00**	**72.31**	31.12	0.00	15.56
# Instances		2,160	746	815	1,026	32	4,779	1,307	46	1,353

Table F.3
Detailed main experimental results on the AbstRCT dataset.

Dataset	Method	$F1_{aci}$				$F1_{ari}$			
		MajorClaim	Evidence	Claim	Macro	Support	Attack	Partial-Attack	Macro
AbstRCT	ST	6.40	72.10	55.24	44.58	39.02	27.64	**27.86**	**31.51**
	UniASA	**7.96**	**73.83**	**62.79**	**48.19**	**41.70**	**33.02**	17.15	30.62
# Instances		93	2,193	993	3,279	1,762	60	238	2,060

Table F.4
Detailed main experimental results on the MTC dataset.

Dataset	Method	$F1_{aci}$			$F1_{ari}$				
		Opponent	Proponent	Macro	Example	Support	Undercut	Rebut	Macro
MTC	ST	**67.11**	79.45	73.28	**10.33**	42.82	37.50	42.66	33.33
	UniASA	63.48	**83.74**	**73.61**	0.0	**48.07**	**43.72**	**46.87**	**34.66**
# Instances		125	451	576	9	281	64	110	464

Table F.5
Detailed main experimental results on the AASD dataset.

Dataset	Method	$F1_{aci}$							$F1_{ari}$					
		Proposal	Means	Observation	Result	Assertion	Description	Macro	Support	Attack	Detail	Sequence	Additional	Macro
AASD	ST	**78.71**	60.42	51.83	**66.57**	**78.57**	0.00	56.02	**58.88**	–	**48.75**	0.00	57.82	41.36
	UniASA	77.85	**61.09**	**64.67**	61.80	77.19	0.00	**57.10**	58.34	–	45.67	0.00	**61.59**	**41.40**
# Instances		110	63	11	74	88	7	353	126	–	129	11	27	293

Appendix G. Results of Binning Test Data Samples by the Number of Gold Components

The number of gold components contained in each data sample is an important data characteristic. Intuitively, the more components a data sample contains, the more likely it is to have a more complex argument structure, and, consequently, the higher the difficulty in prediction. Therefore, we conduct an analysis focused on the model performance across test data samples with different quantities of gold components. Specifically, we divide each test set into three bins based on the number of gold components contained in each data sample, representing high, medium, and low quantities of gold components. The specific division method involves evenly splitting the range of component counts in a data sample into three sub-ranges. For example, in the AAEC dataset, the range of gold component counts in each test data sample is [8, 28]. We then divide this range [8, 28] as evenly as possible into three sub-intervals: [8, 14] (Low), [15, 21] (Medium), and [22, 28] (High).

The analysis results are shown in Table G.1. For the AAEC dataset, UniASA outperforms ST in each bin, and the advantage becomes more pronounced as the number of gold components increases. For the CDCP and AbstRCT datasets, UniASA achieves more significant improvements in the "High" bin. Therefore, we conclude that for E2E-AM, UniASA is more adept than ST when handling data samples with more gold components. Furthermore, for the APE task, UniASA demonstrates its most significant advantage over PIGEON in the "Low" bin, while for the AQE task, UniASA shows the greatest improvement compared to QuadTAG in the "Medium" bin.

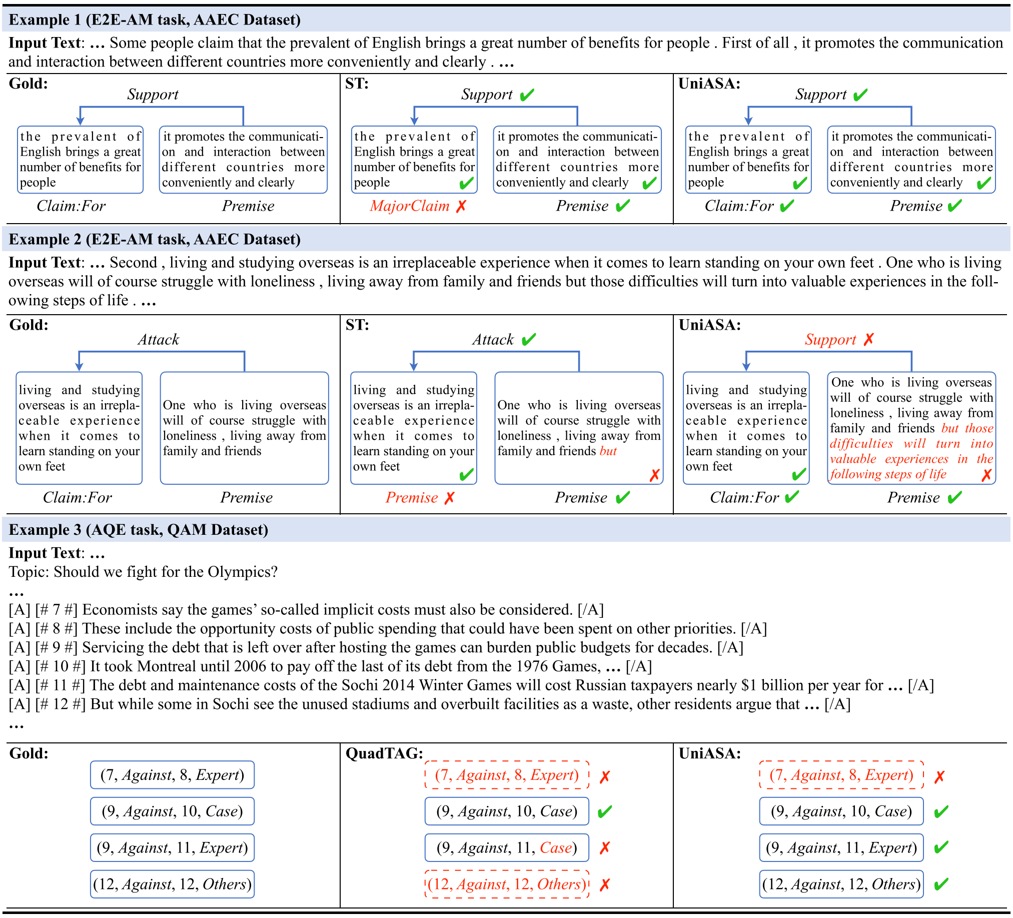
Table G.1
Results of binning test samples by number of gold components. The “Range” column indicates the range of component counts in a data sample within each bin. The “# Test Samples” column indicates the number of test samples contained in each range. Due to the limited data in the MTC and AASD datasets and the absence of official test sets, we do not conduct analysis on these datasets.

Task	Dataset	Bin	Range	# Test Samples	Method	$F1_{aci}$ Micro	$F1_{ari}$ Micro	$F1_{arict}$	$F1_{ape}$	$F1_{aqe}$ Test
E2E-AM	AAEC	Low	[8, 14]	31	ST	70.81	56.29	41.37	–	–
					UniASA	**71.82**	**56.46**	**46.56**	–	–
		Medium	[15, 21]	39	ST	77.07	54.22	47.78	–	–
					UniASA	**79.04**	**55.40**	**49.71**	–	–
		High	[22, 28]	10	ST	79.25	56.47	50.14	–	–
					UniASA	**82.67**	**59.13**	**55.04**	–	–
		All	[8, 28]	80	ST	75.63	55.21	46.57	–	–
					UniASA	**77.60**	**56.33**	**49.83**	–	–
	CDCP	Low	[2, 12]	133	ST	68.35	**34.77**	**27.58**	–	–
					UniASA	**68.43**	32.42	24.73	–	–
		Medium	[13, 22]	13	ST	66.44	**23.75**	**20.37**	–	–
					UniASA	**66.97**	20.51	17.40	–	–
		High	[23, 33]	4	ST	**71.09**	26.51	17.94	–	–
					UniASA	65.67	**41.72**	**24.48**	–	–
		All	[2, 33]	150	ST	**68.26**	**31.45**	**24.96**	–	–
					UniASA	68.16	30.57	23.29	–	–
	AbstRCT	Low	[3, 5]	27	ST	62.00	41.64	38.01	–	–
					UniASA	**71.32**	**44.67**	**45.23**	–	–
		Medium	[6, 8]	52	ST	64.87	37.09	35.09	–	–
					UniASA	**69.03**	**37.39**	**36.67**	–	–
		High	[9, 11]	21	ST	65.44	33.79	31.06	–	–
					UniASA	**70.32**	**38.35**	**37.60**	–	–
		All	[3, 11]	100	ST	64.46	37.04	35.21	–	–
					UniASA	**69.67**	**39.11**	**38.65**	–	–
APE	RR	Low	[0, 13]	399	PIGEON	71.71	–	–	43.82	–
					UniASA	**74.08**	–	–	**45.77**	–
		Medium	[14, 26]	70	PIGEON	74.11	–	–	**43.13**	–
					UniASA	**74.80**	–	–	39.11	–
		High	[27, 39]	5	PIGEON	**75.82**	–	–	**26.04**	–
					UniASA	68.75	–	–	23.93	–
		All	[0, 39]	474	PIGEON	72.75	–	–	42.86	–
					UniASA	**74.10**	–	–	**43.05**	–
AQE	QAM	Low	[1, 8]	195	QuadTAG	–	–	–	–	16.62
					UniASA	–	–	–	–	**17.28**
		Medium	[9, 15]	127	QuadTAG	–	–	–	–	21.14
					UniASA	–	–	–	–	**26.84**
		High	[16, 22]	20	QuadTAG	–	–	–	–	**29.92**
					UniASA	–	–	–	–	29.22
		All	[1, 22]	342	QuadTAG	–	–	–	–	20.60
					UniASA	–	–	–	–	**23.52**

Appendix H. Case Study and Qualitative Error Analysis

We conduct a case study to further illustrate UniASA's pros and cons. Given that the task definitions of the E2E-AM and AQE tasks are more comprehensive and challenging compared to the APE task, we show the case study results on the E2E-AM and AQE tasks. Figure H.1 presents three examples, with the first two from the AAEC dataset and the third from the QAM dataset. For each example, we show the gold label alongside the prediction results of UniASA and the current SOTA methods (ST and QuadTAG).

Example 1. In the first example, ST incorrectly predicts the type of the first argument component as "MajorClaim", whereas its actual type should be "Claim:For". It should be emphasized that in the AAEC dataset, a "Premise" cannot form a relation with a "MajorClaim". This error indicates that ST fails to consider the consistency between component types and argumentative relations. In contrast, our UniASA does take this into account, accurately predicting all the labels for this example.



Example 1 (E2E-AM task, AAEC Dataset)

Input Text: … Some people claim that the prevalent of English brings a great number of benefits for people . First of all , it promotes the communication and interaction between different countries more conveniently and clearly . …

Gold:
Support
the prevalent of English brings a great number of benefits for people
it promotes the communication and interaction between different countries more conveniently and clearly
Claim:For
Premise
ST:
Support ✓
the prevalent of English brings a great number of benefits for people ✓
it promotes the communication and interaction between different countries more conveniently and clearly ✓
MajorClaim ✗
Premise ✓
UniASA:
Support ✓
the prevalent of English brings a great number of benefits for people ✓
it promotes the communication and interaction between different countries more conveniently and clearly ✓
Claim:For ✓
Premise ✓

Example 2 (E2E-AM task, AAEC Dataset)

Input Text: … Second , living and studying overseas is an irreplaceable experience when it comes to learn standing on your own feet . One who is living overseas will of course struggle with loneliness , living away from family and friends but those difficulties will turn into valuable experiences in the following steps of life . …

Gold:
Attack
living and studying overseas is an irreplaceable experience when it comes to learn standing on your own feet
One who is living overseas will of course struggle with loneliness , living away from family and friends
Claim:For
Premise
ST:
Attack ✓
living and studying overseas is an irreplaceable experience when it comes to learn standing on your own feet ✓
One who is living overseas will of course struggle with loneliness , living away from family and friends but ✗
Premise ✗
Premise ✓
UniASA:
Support ✗
living and studying overseas is an irreplaceable experience when it comes to learn standing on your own feet ✓
One who is living overseas will of course struggle with loneliness , living away from family and friends but those difficulties will turn into valuable experiences in the following steps of life ✗
Claim:For ✓
Premise ✓

Example 3 (AQE task, QAM Dataset)

Input Text: …
Topic: Should we fight for the Olympics?
…
[A] [# 7 #] Economists say the games' so-called implicit costs must also be considered. [/A]
[A] [# 8 #] These include the opportunity costs of public spending that could have been spent on other priorities. [/A]
[A] [# 9 #] Servicing the debt that is left over after hosting the games can burden public budgets for decades. [/A]
[A] [# 10 #] It took Montreal until 2006 to pay off the last of its debt from the 1976 Games, … [/A]
[A] [# 11 #] The debt and maintenance costs of the Sochi 2014 Winter Games will cost Russian taxpayers nearly $1 billion per year for … [/A]
[A] [# 12 #] But while some in Sochi see the unused stadiums and overbuilt facilities as a waste, other residents argue that … [/A]
…

Gold:	QuadTAG:	UniASA:
(7, *Against*, 8, *Expert*)	(7, *Against*, 8, *Expert*) ✗	(7, *Against*, 8, *Expert*) ✗
(9, *Against*, 10, *Case*)	(9, *Against*, 10, *Case*) ✓	(9, *Against*, 10, *Case*) ✓
(9, *Against*, 11, *Expert*)	(9, *Against*, 11, *Case*) ✗	(9, *Against*, 11, *Expert*) ✓
(12, *Against*, 12, *Others*)	(12, *Against*, 12, *Others*) ✗	(12, *Against*, 12, *Others*) ✓

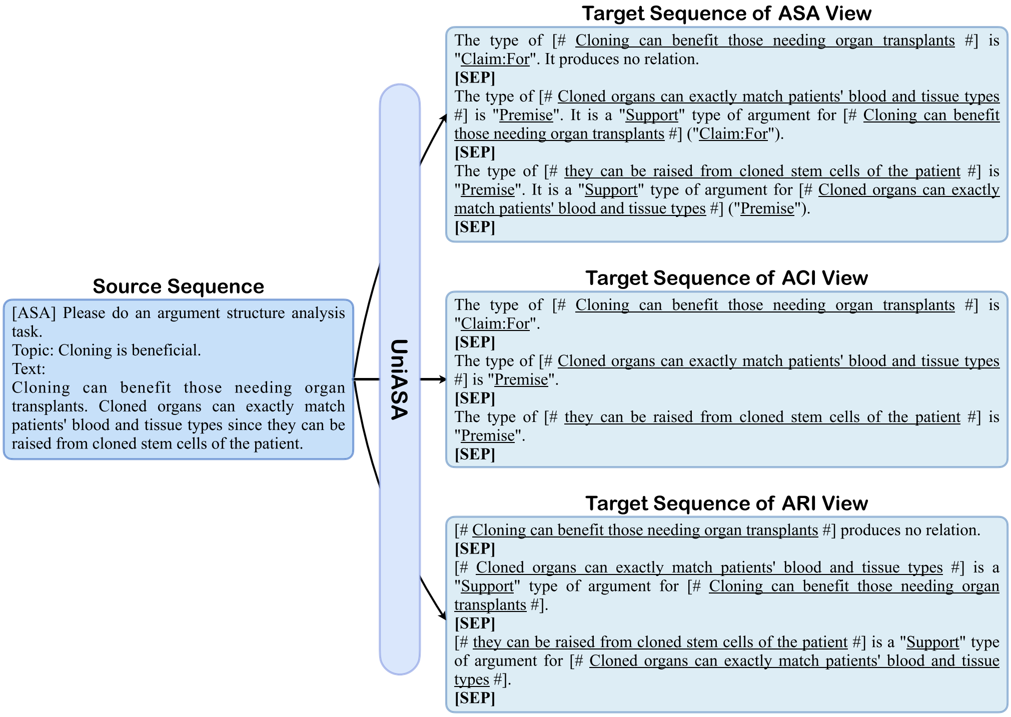
Figure H.1
Case study and error analysis. The incorrectly predicted parts are highlighted in red.

Example 2. The second example contains an "Attack" type of relation from a "Premise" to a "Claim:For". ST incorrectly predicted both the type of the first component and the span of the second component. Note that the span mistake makes the semantics of the second component incomplete. UniASA does not achieve satisfactory results in this example either. It also incorrectly predicts the span of the second component, which directly leads to the incorrect relation prediction that misidentifies "Attack" as "Support". However, it should be noted that even though the span prediction for the second component by UniASA is incorrect, compared to ST, the span predicted by UniASA is semantically complete. Additionally, based on this erroneously predicted span, the prediction of the "Support" relation is logical, which also demonstrates UniASA's adherence to consistency. Nevertheless, there is still room for improvement for UniASA in the segmentation of component spans.

Example 3. The third example is from the QAM dataset, where the prediction target is a series of argument quadruplets, including a claim, the claim's stance, evidence supporting the claim, and the type of evidence. This example consists of four quadruplets. It is observed that the current SOTA model, QuadTAG, shows poor performance. It correctly predicts only one quadruplet, misclassifies the evidence type in another, and completely fails to predict the other two quadruplets. In contrast, UniASA successfully predicts three out of the four quadruplets, only missing one.

Appendix I. An Example of Multi-view Learning

Figure I.1 illustrates a simplified example of multi-view learning, corresponding to the example shown in Figure 1.


Source Sequence
UniASA
Target Sequence of ASA View
Target Sequence of ACI View
Target Sequence of ARI View

[ASA] Please do an argument structure analysis task.
Topic: Cloning is beneficial.
Text:
Cloning can benefit those needing organ transplants. Cloned organs can exactly match patients' blood and tissue types since they can be raised from cloned stem cells of the patient.

The type of [# Cloning can benefit those needing organ transplants #] is "Claim:For". It produces no relation.
[SEP]
The type of [# Cloned organs can exactly match patients' blood and tissue types #] is "Premise". It is a "Support" type of argument for [# Cloning can benefit those needing organ transplants #] ("Claim:For").
[SEP]
The type of [# they can be raised from cloned stem cells of the patient #] is "Premise". It is a "Support" type of argument for [# Cloned organs can exactly match patients' blood and tissue types #] ("Premise").
[SEP]

The type of [# Cloning can benefit those needing organ transplants #] is "Claim:For".
[SEP]
The type of [# Cloned organs can exactly match patients' blood and tissue types #] is "Premise".
[SEP]
The type of [# they can be raised from cloned stem cells of the patient #] is "Premise".
[SEP]

[# Cloning can benefit those needing organ transplants #] produces no relation.
[SEP]
[# Cloned organs can exactly match patients' blood and tissue types #] is a "Support" type of argument for [# Cloning can benefit those needing organ transplants #].
[SEP]
[# they can be raised from cloned stem cells of the patient #] is a "Support" type of argument for [# Cloned organs can exactly match patients' blood and tissue types #].
[SEP]

Figure I.1
An example of multi-view learning. Through the multi-view learning strategy, a single source sequence can generate three target sequences from different views. This figure illustrates a simplified example from the AAEC dataset of the E2E-AM task, corresponding to the example shown in Figure 1.

Appendix J. More Ablation Analysis of Multi-view Learning

To further analyze the effectiveness of multi-view learning, we present our UniASA's performance across different views in Table J.1. Our experiments show that the multi-view learning strategy with voting-based aggregation ("UniASA All") outperforms the baseline without multi-view learning ("UniASA *w/o mv*") on both E2E-AM and AQE tasks. For the APE task, both approaches achieve comparable performance. Furthermore, aggregating predictions through voting across all views generally yields better results than individual view predictions.

Table J.1
More ablation analysis of multi-view learning. The methods shown here are based on t5-base or long-t5-tglobal-base models, as discussed in Section 5. "ACI", "ARI", and "ASA" represent the results of individual views, while "All" represents the results after aggregating three views by voting. For the AQE task, since complete quadruplet extraction results cannot be obtained from individual "ACI" and "ARI" views, we only consider the "ASA" view.

Task	Dataset	Method	View	F1$_{aci}$ Micro	F1$_{ari}$ Micro	F1$_{arict}$	F1$_{ape}$	F1$_{aqe}$ Test
E2E-AM	AAEC	UniASA	ACI	**77.68**	–	–	–	–
			ARI	–	56.21	–	–	–
			ASA	77.49	56.13	49.46	–	–
			All	77.60	**56.33**	**49.83**	–	–
		UniASA *w/o mv*	ASA	76.24	55.37	48.59	–	–
	CDCP	UniASA	ACI	67.31	–	–	–	–
			ARI	–	30.48	–	–	–
			ASA	67.77	**30.80**	22.76	–	–
			All	**68.16**	30.57	**23.29**	–	–
		UniASA *w/o mv*	ASA	67.64	28.37	21.86	–	–
	AbstRCT	UniASA	ACI	69.35	–	–	–	–
			ARI	–	38.97	–	–	–
			ASA	68.69	**39.38**	38.03	–	–
			All	**69.67**	39.11	**38.65**	–	–
		UniASA *w/o mv*	ASA	68.56	35.95	34.84	–	–
	MTC	UniASA	ACI	79.67	–	–	–	–
			ARI	–	45.54	–	–	–
			ASA	79.74	45.63	44.26	–	–
			All	**79.88**	**46.76**	**44.44**	–	–
		UniASA *w/o mv*	ASA	78.90	45.68	42.68	–	–
	AASD	UniASA	ACI	68.41	–	–	–	–
			ARI	–	**49.63**	–	–	–
			ASA	**68.86**	49.18	43.06	–	–
			All	68.84	49.54	**44.78**	–	–
		UniASA *w/o mv*	ASA	68.17	48.23	42.32	–	–
APE	RR	UniASA	ACI	74.08	–	–	–	–
			ARI	–	–	–	42.99	–
			ASA	74.03	–	–	42.53	–
			All	**74.10**	–	–	43.05	–
		UniASA *w/o mv*	ASA	73.77	–	–	**43.53**	–
AQE	QAM	UniASA	ASA	–	–	–	–	23.35
			All	–	–	–	–	**23.52**
		UniASA *w/o mv*	ASA	–	–	–	–	22.22

Appendix K. Statistics of the Argument Structure Analysis Datasets

Table K.1
Statistics of the argument structure analysis datasets. For the APE task, "# Relations" indicates the count of argument pairs, while for the AQE task, it represents the number of argument quadruplets.

Task	Dataset	# Instance	# Components	# Relations
E2E-AM	AAEC	402	6,089	3,832
	CDCP	731	4,779	1,353
	AbstRCT	500	3,279	2,060
	MTC	112	576	464
	AASD	60	353	293
APE	RR	4,764	41,230	19,117
AQE	QAM	801	–	25,563

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China 62176076, Natural Science Foundation of GuangDong 2023A1515012922, Shenzhen Foundational Research Funding JCYJ20220818102415032, Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies 2022B1212010005, and The Major Key Project of PCL PCL2023A09.

References

Accuosto, Pablo, Mariana Neves, and Horacio Saggion. 2021. Argumentation mining in scientific literature: From computational linguistics to biomedicine. In *Proceedings of the 11th International Workshop on Bibliometric-enhanced Information Retrieval co-located with 43rd European Conference on Information Retrieval (ECIR 2021)*, pages 20–36.

Bao, Jianzhu, Chuang Fan, Jipeng Wu, Yixue Dang, Jiachen Du, and Ruifeng Xu. 2021a. 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, ACL/IJCNLP 2021*, pages 6354–6364. https://doi.org/10.18653/v1/2021.acl-long.497

Bao, Jianzhu, Yuhang He, Yang Sun, Bin Liang, Jiachen Du, Bing Qin, Min Yang, and Ruifeng Xu. 2022a. 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, EMNLP 2022*, pages 10437–10449. https://doi.org/10.18653/v1/2022.emnlp-main.713

Bao, Jianzhu, Bin Liang, Jingyi Sun, Yice Zhang, Min Yang, and Ruifeng Xu. 2021b. Argument pair extraction with mutual guidance and inter-sentence relation graph. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 3923–3934. https://doi.org/10.18653/v1/2021.emnlp-main.319

Bao, Jianzhu, Jingyi Sun, Qinglin Zhu, and Ruifeng Xu. 2022b. Have my arguments been replied to? Argument pair extraction as machine reading comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022*, pages 29–35. https://doi.org/10.18653/v1/2022.acl-short.4

Beardsley, Monroe C. 1950. *Thinking Straight*. Prentice-Hall.

Beltagy, Iz, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, pages 3613–3618. https://doi.org/10.18653/v1/D19-1371

Chen, Zhi, Lu Chen, Bei Chen, Libo Qin, Yuncong Liu, Su Zhu, Jian-Guang Lou, and Kai Yu. 2022. Unidu: Towards a unified generative dialogue understanding framework. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGDIAL 2022,* pages 442–455. https://doi.org/10.18653/v1/2022.sigdial-1.43

Cheng, Liying, Lidong Bing, Qian Yu, Wei Lu, and Luo Si. 2020. APE: Argument pair extraction from peer review and rebuttal via multi-task learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 7000–7011. https://doi.org/10.18653/v1/2020.emnlp-main.569

Cheng, Liying, Tianyu Wu, Lidong Bing, and Luo Si. 2021. Argument pair extraction via attention-guided multi-layer multi-cross encoding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021*, pages 6341–6353. https://doi.org/10.18653/v1/2021.acl-long.496

Chernodub, Artem N., Oleksiy Oliynyk, Philipp Heidenreich, Alexander Bondarenko, Matthias Hagen, Chris Biemann, and Alexander Panchenko. 2019. TARGER: Neural argument mining at your fingertips. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 195–200. https://doi.org/10.18653/v1/P19-3031

Cocarascu, Oana and Francesca Toni. 2017. Identifying attack and support argumentative relations using deep learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 1374–1379. https://doi.org/10.18653/v1/D17-1144

Dettmers, Tim, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*.

Eger, Steffen, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural end-to-end learning for computational argumentation mining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, pages 11–22. https://doi.org/10.18653/v1/P17-1002

Elaraby, Mohamed, Yang Zhong, and Diane J. Litman. 2023. Towards argument-aware abstractive summarization of long legal opinions with summary reranking. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7601–7612. https://doi.org/10.18653/v1/2023.findings-acl.481

Freeman, James B. 2011. *Argument Structure: Representation and Theory*. Springer. https://doi.org/10.1007/978-94-007-0357-5

Gao, Tianhao, Jun Fang, Hanyu Liu, Zhiyuan Liu, Chao Liu, Pengzhang Liu, Yongjun Bao, and Weipeng Yan. 2022. LEGO-ABSA: A prompt-based task assemblable unified generative framework for multi-task aspect-based sentiment analysis. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022*, pages 7002–7012.

Guo, Jia, Liying Cheng, Wenxuan Zhang, Stanley Kok, Xin Li, and Lidong Bing. 2023. AQE: Argument quadruplet extraction via a quad-tagging augmented generative approach. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 932–946. https://doi.org/10.18653/v1/2023.findings-acl.59

Guo, Mandy, Joshua Ainslie, David C. Uthus, Santiago Ontañón, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. LongT5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736. https://doi.org/10.18653/v1/2022.findings-naacl.55

He, Wanwei, Yinpei Dai, Min Yang, Jian Sun, Fei Huang, Luo Si, and Yongbin Li. 2022. Unified dialog model pre-training for task-oriented dialog understanding and generation. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 187–200. https://doi.org/10.1145/3477495.3532069

Hu, Guimin, Ting-En Lin, Yi Zhao, Guangming Lu, Yuchuan Wu, and Yongbin Li. 2022. UniMSE: Towards unified multimodal sentiment analysis and emotion recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*, pages 7837–7851. https://doi.org/10.18653/v1/2022.emnlp-main.534

Hua, Xinyu, Zhe Hu, and Lu Wang. 2019. Argument generation with retrieval, planning, and realization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 2661–2672. https://doi.org/10.18653/v1/P19-1255

Ji, Lu, Zhongyu Wei, Jing Li, Qi Zhang, and Xuanjing Huang. 2021. Discrete argument representation learning for interactive argument pair identification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021*, pages 5467–5478. https://doi.org/10.18653/v1/2021.naacl-main.431

Jo, Yohan, Seojin Bang, Chris Reed, and Eduard H. Hovy. 2021. Classifying argumentative relations using logical mechanisms and argumentation schemes. *Transactions of the Association for Computational Linguistics*, 9:721–739. https://doi.org/10.1162/tacl_a_00394

Kuribayashi, Tatsuki, 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 Conference of the Association for Computational Linguistics, ACL 2019*, pages 4691–4698. https://doi.org/10.18653/v1/P19-1464

Lauscher, Anne, Henning Wachsmuth, Iryna Gurevych, and Goran Glavas. 2022. Scientia potentia est—On the role of knowledge in computational argumentation. *Transactions of the Association for Computational Linguistics*, 10:1392–1422. https://doi.org/10.1162/tacl_a_00525

Lawrence, John and Chris Reed. 2019. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818. https://doi.org/10.1162/coli_a_00364

Li, Wei, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. 2021. UNIMO: Towards unified-modal understanding and generation via cross-modal contrastive learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021*, pages 2592–2607. https://doi.org/10.18653/v1/2021.acl-long.202

Li, Zaijing, Ting-En Lin, Yuchuan Wu, Meng Liu, Fengxiao Tang, Ming Zhao, and Yongbin Li. 2023. UniSA: Unified generative framework for sentiment analysis. In *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023*, pages 6132–6142. https://doi.org/10.1145/3581783.3612336

Liu, Boyang, Viktor Schlegel, Paul Thompson, Riza Theresa Batista-Navarro, and Sophia Ananiadou. 2023. Global information-aware argument mining based on a top-down multi-turn QA model. *Information Processing & Management*, 60(5):103445. https://doi.org/10.1016/j.ipm.2023.103445

Liu, Xiaodong, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 4487–4496. https://doi.org/10.18653/v1/P19-1441

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lu, Jinghui, Rui Zhao, Brian Mac Namee, and Fei Tan. 2023. PUnifiedNER: A prompting-based unified NER system for diverse datasets. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023*, pages 13327–13335. https://doi.org/10.1609/aaai.v37i11.26564

Lu, Yaojie, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022*, pages 5755–5772. https://doi.org/10.18653/v1/2022.acl-long.395

Mayer, Tobias, Elena Cabrio, and Serena Villata. 2020. Transformer-based argument mining for healthcare applications. In *ECAI 2020 - 24th European Conference on Artificial Intelligence*, pages 2108–2115. https://doi.org/10.3233/FAIA200334

Mestre, Rafael, Razvan Milicin, Stuart E. Middleton, Matt Ryan, Jiatong Zhu, and Timothy J. Norman. 2021. M-Arg: Multimodal argument mining dataset for political debates with audio and transcripts. In *Proceedings of the 8th Workshop on Argument Mining*, pages 78–88. https://doi.org/10.18653/v1/2021.argmining-1.8

Morio, Gaku, 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, ACL 2020*, pages 3259–3266. `https://doi.org/10.18653/v1/2020.acl-main.298`

Morio, Gaku, 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. `https://doi.org/10.1162/tacl_a_00481`

Palau, Raquel Mochales and Marie-Francine Moens. 2009. Argumentation mining: the detection, classification and structure of arguments in text. In the *12th International Conference on Artificial Intelligence and Law, Proceedings of the Conference*, pages 98–107. `https://doi.org/10.1145/1568234.1568246`

Park, Joonsuk and Claire Cardie. 2018. A corpus of erulemaking user comments for measuring evaluability of arguments. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018*.

Peldszus, Andreas. 2014. Towards segment-based recognition of argumentation structure in short texts. In *Proceedings of the First Workshop on Argument Mining, hosted by the 52nd Annual Meeting of the Association for Computational Linguistics, ArgMining@ACL 2014*, pages 88–97. `https://doi.org/10.3115/v1/W14-2112`

Peldszus, Andreas and Manfred Stede. 2015. An annotated corpus of argumentative microtexts. In *1st European Conference on Argumentation*, pages 801–815.

Persing, Isaac and Vincent Ng. 2016. End-to-end argumentation mining in student essays. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1384–1394. `https://doi.org/10.18653/v1/N16-1164`

Potash, Peter, 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, EMNLP 2017*, pages 1364–1373. `https://doi.org/10.18653/v1/D17-1143`

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21:140:1–140:67.

Saadat-Yazdi, Ameer, Jeff Z. Pan, and Nadin Kökciyan. 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, EACL 2023*, pages 2476–2487. `https://doi.org/10.18653/v1/2023.eacl-main.182`

Song, Wei, Ziyao Song, Lizhen Liu, and Ruiji Fu. 2020. Hierarchical multi-task learning for organization evaluation of argumentative student essays. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3875–3881. `https://doi.org/10.24963/ijcai.2020/536`

Stab, Christian and Iryna Gurevych. 2014. Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pages 46–56. `https://doi.org/10.3115/v1/D14-1006`

Stab, Christian and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659. `https://doi.org/10.1162/COLI_a_00295`

Sun, Yang, 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. `https://doi.org/10.18653/v1/2022.findings-emnlp.264`

Sun, Yang, Bin Liang, Jianzhu Bao, Yice Zhang, Geng Tu, Min Yang, and Ruifeng Xu. 2023. Probing graph decomposition for argument pair extraction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13075–13088. `https://doi.org/10.18653/v1/2023.findings-acl.827`

Thomas, Stephen Naylor. 1973. *Practical Reasoning in Natural Language*. Prentice-Hall.

Toulmin, Stephen. 1958. *The Uses of Argument*. Cambridge University Press.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Trautmann, Dietrich, Johannes Daxenberger, Christian Stab, Hinrich Schütze, and Iryna Gurevych. 2020. Fine-grained argument unit recognition and classification. In the *Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 9048–9056. https://doi.org/10.1609/aaai.v34i05.6438

Van Eemeren, Frans H., Robert Grootendorst, and Rob Grootendorst. 2004. *A Systematic Theory of Argumentation: The Pragma-dialectical Approach*. Cambridge University Press. https://doi.org/10.1017/CBO9780511616389

Walton, Douglas. 1996. *Argument Structure: A Pragmatic Theory*. University of Toronto Press. https://doi.org/10.3138/9781487574475

Wang, Hao, Zhen Huang, Yong Dou, and Yu Hong. 2020. Argumentation mining on essays at multi scales. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020*, pages 5480–5493. https://doi.org/10.18653/v1/2020.coling-main.478

Weber, Florian, Thiemo Wambsganss, Seyed Parsa Neshaei, and Matthias Söllner. 2023. Structured persuasive writing support in legal education: A model and tool for German legal case solutions. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2296–2313. https://doi.org/10.18653/v1/2023.findings-acl.145

Yan, Hang, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. 2021a. A unified generative framework for 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, ACL/IJCNLP 2021*, pages 2416–2429. https://doi.org/10.18653/v1/2021.acl-long.188

Yan, Hang, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021b. A unified generative framework for various NER subtasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021*, pages 5808–5822. https://doi.org/10.18653/v1/2021.acl-long.451

Ye, Yuxiao and Simone Teufel. 2021. End-to-end argument mining as biaffine dependency parsing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2021*, pages 669–678. https://doi.org/10.18653/v1/2021.eacl-main.55

Yuan, Jian, Zhongyu Wei, Yixu Gao, Wei Chen, Yun Song, Donghua Zhao, Jinglei Ma, Zhen Hu, Shaokun Zou, Donghai Li, and Xuanjing Huang. 2021a. Overview of SMP-CAIL2020-Argmine: The interactive argument-pair extraction in judgement document challenge. *Data Intelligence*, 3(2):287–307. https://doi.org/10.1162/dint_a_00094

Yuan, Jian, Zhongyu Wei, Donghua Zhao, Qi Zhang, and Changjian Jiang. 2021b. Leveraging argumentation knowledge graph for interactive argument pair identification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2310–2319. https://doi.org/10.18653/v1/2021.findings-acl.203

Zhang, Xinpeng, Ming Tan, Jingfan Zhang, and Wei Zhu. 2023. NAG-NER: A unified non-autoregressive generation framework for various NER tasks. In *Proceedings of the the 61st Annual Meeting of the Association for Computational Linguistics: Industry Track, ACL 2023*, pages 676–686. https://doi.org/10.18653/v1/2023.acl-industry.65

Zhang, Zhengkun, Xiaojun Meng, Yasheng Wang, Xin Jiang, Qun Liu, and Zhenglu Yang. 2022. UniMS: A unified framework for multimodal summarization with knowledge distillation. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022*, pages 11757–11764. https://doi.org/10.1609/aaai.v36i10.21431

Zhu, Xiaofei, Yidan Liu, Zhuo Chen, Xu Chen, Jiafeng Guo, and Stefan Dietze. 2023. A mutually enhanced multi-scale relation-aware graph convolutional network for argument pair extraction. *Journal of Intelligent Information Systems*, pages 1–20. https://doi.org/10.1007/s10844-023-00826-9