Learning First-Order Logic Rules for Argumentation Mining

Yang Sun[1,2*], **Guanrong Chen**[1,5*], **Hamid Alinejad-Rokny**[6], **Jianzhu Bao**[1,2], **Yuqi Huang**[1], **Bin Liang**[1,4], **Kam-Fai Wong**,[4] **Min Yang**[3†] and **Ruifeng Xu**[1,2,5†]
[1] Harbin Institute of Technology, Shenzhen, China [2] Peng Cheng Laboratory, Shenzhen, China
[3] SIAT, Chinese Academy of Sciences, Shenzhen, China [4] The Chinese University of Hong Kong
[5] Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies
[6] School of Biomedical Engineering, UNSW Sydney
yang.sun@stu.hit.edu.cn, 23S051030@stu.hit.edu.cn

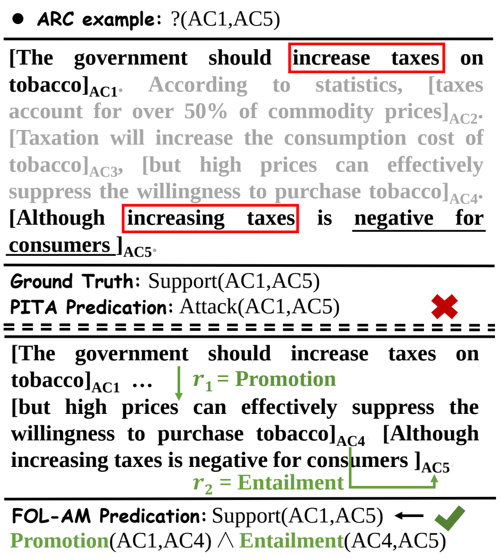
Abstract

Argumentation Mining (AM) aims to extract argumentative structures from texts by identifying argumentation components (ACs) and their argumentative relations (ARs). While previous works focus on representation learning to encode ACs and AC pairs, they fail to explicitly model the underlying reasoning patterns of AM, resulting in limited interpretability. This paper proposes a novel First-Order Logic reasoning framework for AM (FOL-AM), designed to explicitly capture logical reasoning paths within argumentative texts. By interpreting multiple AM subtasks as a unified relation query task modeled using FOL rules, FOL-AM facilitates multi-hop relational reasoning and enhances interpretability. The framework supports two flexible implementations: a fine-tuned approach to leverage task-specific learning, and a prompt-based method utilizing large language models to harness their generalization capabilities. Extensive experiments on two AM benchmarks demonstrate that FOL-AM outperforms strong baselines while significantly improving explainability.

1 Introduction

Argumentation Mining (AM) focuses on extracting argumentation structures from texts, which has garnered significant attention in recent years (Lawrence and Reed, 2019; Schaefer and Stede, 2021; Vecchi et al., 2021). AM enables automation of human argumentation logic and has been shown to provide support across various domains, such as decision-making (Vassiliades et al., 2021; Walker et al., 2018), writing support (Wambsganss and Rietsche, 2019; Wambsganss et al., 2020), and argument generation (Khatib et al., 2021; Hua et al., 2019; Slonim et al., 2021).

Given an argumentative text, an Argumentation Mining (AM) system must identify argumentation components (ACs) and their corresponding argumentative relations (ARs). AM generally consists of three key subtasks (Morio et al., 2022a; Stab and Gurevych, 2017a): (i) *Argumentation Component Segmentation* (ACS), which identifies and extracts the boundaries of ACs within the input text; (ii) *Argumentation Component Type Classification* (ACTC), which categorizes ACs into predefined types (e.g., *Claim* and *Premise*); and (iii) *Argumentative Relation Classification* (ARC), which determines the type of relation (e.g., *Support* or *Attack*) between pairs of ACs. Following previous works (Potash et al., 2017a; Kuribayashi et al., 2019a), we assume that the ACS step has already been performed, meaning that ACs are pre-segmented. Thus, we focus on addressing the ACTC and ARC subtasks.


● **ARC example:** ?(AC1,AC5)

[The government should increase taxes on tobacco]$_{AC1}$. According to statistics, [taxes account for over 50% of commodity prices]$_{AC2}$. [Taxation will increase the consumption cost of tobacco]$_{AC3}$, [but high prices can effectively suppress the willingness to purchase tobacco]$_{AC4}$. **[Although increasing taxes is negative for consumers]$_{AC5}$.**

Ground Truth: Support(AC1,AC5)
PITA Predication: Attack(AC1,AC5) ✗

[The government should increase taxes on tobacco]$_{AC1}$ … r_1 = Promotion
[but high prices can effectively suppress the willingness to purchase tobacco]$_{AC4}$ [Although increasing taxes is negative for consumers]$_{AC5}$
r_2 = Entailment

FOL-AM Predication: Support(AC1,AC5) ← ✓
Promotion(AC1,AC4) $\wedge$ Entailment(AC4,AC5)

Figure 1: Illustration of ARC subtask, comparing the results of PITA (previous state-of-the-art model) and FOL-AM (ours).

Previous works (Kuribayashi et al., 2019b; Bao et al., 2021a; Morio et al., 2022b; Sun et al., 2024) primarily focus on representation learning to encode ACs and AC pairs for AM. For instance,

* Equal Contribution.
† Min Yang and Ruifeng Xu are corresponding authors.

Kuribayashi et al. (2019b) enhances AC representations using discourse markers, while Bao et al. (2021a) employs a transition-based network for AC and AC pair modeling. Morio et al. (2022b) leverages biaffine functions to encode relational representations, and Sun et al. (2024) incorporates task interactions into AC and AC pair representations. However, these methods fail to explicitly model the reasoning patterns underlying AM, potentially leading to sub-optimal performance. Moreover, they lack interpretability, as they rely on black-box neural networks.

For example, as shown in Figure 1, the state-of-the-art (SOTA) model PITA misclassifies the relation between AC1 and AC5 as *Attack* due to their shared contextual cue *increase taxes*, whereas AC5 actually expresses a negative stance toward the topic. In contrast, humans can deduce the correct relation using logical reasoning, recognizing that "if AC1 promotes AC4, and AC4 entails AC5, then AC1 and AC5 are in a *Support* relation."

In this paper, we propose a novel first-order logic reasoning framework for multi-task AM (FOL-AM). As indicated in Figure 1, our method aligns with human reasoning by introducing interpretable FOL rules derived from the argumentative text for AM. For example, we may use the following logic rule to identify the relation between AC pair $AC1$ and $AC5$ as "Support": Support(AC1, AC5) $\leftarrow$ Promotion [1](AC1, AC4) $\wedge$ Entailment(AC4, AC5). This logic reasoning framework has two clear benefits over previous approaches: First, it naturally captures multi-hop relations among ACs owing to the compositional structure of the reasoning chain. Second, it improves interpretability as the reasoning processes are visible.

Specifically, we uniformly transform ACTC and ARC subtasks as relation query $q(x_1, x_{L+1})$ associated with an AC pair (x_1, x_{L+1}) within the argumentative text. The query is modeled as the FOL rule to conduct logical deduction: $q(x_1, x_{L+1}) \leftarrow r_1(x_1, x_2) \wedge \cdots \wedge r_L(x_L, x_{L+1})$. The body of the rule (on the left) is formed by attentively selecting proper ACs x_i and predicates r_i upon given argumentative text. We implement the FOL-AM framework in two settings: a fine-tuned method and a prompt-based method using LLMs. Intuitively, the former method can learn the nuanced patterns specific to the AM task directly from the dataset, while the latter can leverage the remarkable performance of the LLMs, which is supposed to have better generalization in various corpora.

In summary, our contributions are as follows: (i) We introduce the first FOL-based reasoning framework for multi-task AM; (ii) We propose a unified FOL rule-based approach to model reasoning patterns in AM; (iii) We implement both fine-tuned and LLM prompt-based methods, demonstrating the flexibility of our framework; (iv) Extensive experiments on two AM datasets showcase the superiority of FOL-AM.

2 Background

Task Definition Building on previous works (Potash et al., 2017a; Morio et al., 2020a), we assume that the ACS subtask has been completed, meaning the spans of ACs are given. Given an argumentative text $W = \{w_1, w_2, \cdots, w_n\}$ consisting of n tokens and m ACs $X = \{x_1, x_2, \cdots, x_m\}$, with the start and end word indexes (b_i, e_i) of each AC x_i, the objective is twofold: first, to classify each AC's type $y_i \in Y_{\text{ACTC}}$ (e.g., *Claim* or *Premise*), and second, to determine the argumentative relation (AR) type $y_{(i,j)} \in Y_{\text{ARC}}$ (e.g., *Support*, *Attack*, or *No-Relation*) between two ACs x_i and x_j.

FOL Rule In this work, we define a L-hop FOL rule that has the conjunctive form as follows:

$$q(a_1, a_{L+1}) \leftarrow r_1(a_1, a_2) \wedge r_2(a_2, a_3) \cdots \wedge r_L(a_L, a_{L+1}) \quad (1)$$

where a_i denotes an argument, r_i denotes a predicate, i.e., a relation between a_i and a_{i+1}. Each $r_i(a_i, a_{i+1})$ is a logic atom and an atom is composed of a predicate r_i and its arguments a_i and a_{i+1}. The left side of $\leftarrow$ represents the head atom, while the right side constitutes a rule body. L denotes the number of predicates in the rule body. In this way, the body rule suggests a reasoning path to deduce the relation query $q(a_1, a_{L+1})$ in the head atom.

The ACTC and ARC have different task objectives that hinder the direct incorporation of FOL rules into a multi-task AM. Concretely, ACTC classifies individual ACs, while ARC predicts the relation type between AC pairs. To bridge the gap between multi-task AM and FOL rule, we reformulate the ACTC and ARC subtasks into a unified relation query task driven by the FOL rule. For ACTC, we define the relation query $q(x_i, x_W)$ as the head atom of a FOL rule that predicts the AC

[1]These phrases serve as human-interpretable descriptions of the abstract logical predicates induced via LLM prompting, as described in Section 3.3.

type y_{x_i} of the AC x_i. Similarly, for ARC, the relation query $q(x_i, x_j)$ is defined as the head atom of a FOL rule that predicts the AR type $y_{(x_i,x_j)}$ for AC pair (x_i, x_j). Here, we introduce the argumentative text input W as a special AC x_W. This is because the type of an AC is determined by its context and its relations with other ACs (Stab and Gurevych, 2014).

3 Methodologies

We propose FOL-AM, a First-Order Logic (FOL) reasoning framework designed to explicitly uncover the reasoning patterns in Argumentation Mining (AM). FOL-AM consists of two key components: (i) an *FOL Term Constructor*, which learns representations of Argumentation Components (ACs) and predicates; and (ii) an *FOL Rule Generator*, which utilizes these representations to instantiate logical atoms. The rule bodies are then constructed by selecting relevant atoms, and subsequently evaluated to infer the relation query in the head atom.

3.1 FOL Term Constructor

AC Representation Learning We define the ACs as the argument candidates within the FOL rules, as the relation query in the head atom operates on AC pairs. Given an argumentative text W with AC set X, we employ a Longformer (Beltagy et al., 2020) to model the representations of the text W and ACs. Specifically, the text W is passed through Longformer to obtain contextual representations H^W. Then the whole text representation $\mathbf{h}^W$ is then derived by applying mean-pooling over H^W. For each AC x_i, its representation $\mathbf{h}_i$ is computed by mean-pooling over the tokens in the span (b_i, e_i), i.e., $\mathbf{h}_i = \frac{1}{e_i - b_i + 1} \sum_{k=b_i}^{e_i} H_k^W$.

Predicate Generation We assume that there are M atomic predicates with indecomposable semantics, represented by a predicate set $\mathcal{R} = \{r_i\}_{i=1}^{M}$. Each predicate is assigned a learnable vector to represent its semantics, with $U \in \mathbb{R}^{M \times d}$ denoting the embedding matrix for the M predicates.

3.2 FOL Rule Generator

The FOL Rule Generator aims to conduct complex FOL reasoning by first generating multiple probable rule bodies and then ranking each rule body to derive the final FOL rule.

3.2.1 Rules Generation

Given the AC set X and predicate set $\mathcal{R}$, the l-hop FOL rule $q(x_1, x_{l+1}) \leftarrow r_1(x_1, a_2) \wedge \cdots \wedge r_l(a_l, x_{l+1})$ [2] consists of atoms $r_t(a_t, a_{t+1})$ defined over arguments a_t and predicates r_t to derive the relation query $q(x_1, x_{l+1})$. Generating the FOL rule involves three steps: (1) AC autoregression learning selects relevant ACs $\{x_2, \cdots, x_l\}$ to instantiate the bridging arguments $\{a_2, \cdots, a_l\}$ in the rule body. (2) Predicate autoregressive learning sequentially generates predicate r_t for each atom in the rule body, ensuring logical consistency across steps. (3) Body atom generation integrates the selected ACs and predicates to form the body atoms. These steps work together to generate multiple probable rule bodies.

AC Autoregression Learning We first apply AC autoregression learning to select ACs from which one may be chosen as a bridging argument $a_t (1 < t \leq l)$ of a rule body. The first and last arguments (a_1, a_{l+1}) are referred to as (x_1, x_{l+1}) based on the head atom $q(x_1, x_{l+1})$. Specifically, we calculate a score A_t^l for each AC using the previously instantiated argument x_{t-1}. The score indicates the probability of selecting an AC as the instantiation of the argument a_t from the AC set X. For example, the score for x_t is:

$$A_t^l = \frac{\exp(\mathbf{W}^A[\mathbf{h}_{t-1}; \mathbf{h}_t])}{\sum_{k=1}^{m} \exp(\mathbf{W}^A[\mathbf{h}_{t-1}; \mathbf{h}_k])} \tag{2}$$

where W^A is a learnable weight matrix. $\mathbf{h}_{t-1}$ is the AC representation of instantiated argument x_{t-1}. $[;]$ represents concatenation. Instead of selecting only the AC with the highest score, we choose the top-K ACs $\{x_t^1, \ldots, x_t^K\} \subseteq X$ as the possible instantiation of the argument a_t according to A_t^l. This is because the local optimum derived from autoregression learning may not be the global optimum, as demonstrated in Section 4.6.

Predicate Autoregressive Learning Then, we learn a predicate association that indicates which predicates are likely to be generated for constructing a rule body. To do this, we denote $\mathbf{u}^l \in \mathbb{R}^D$ as the query vector for the l-hop reasoning rule and use predicate autoregressive learning to generate a probability vector $s_t^l \in \mathbb{R}^M$. This probability vector represents the distribution of the predicate in the t-th atom over the predicate set $\mathcal{R}$:

$$\mathbf{s}_t^l = \mathrm{softmax}((W_t^l \mathbf{u}_t^l)^T (W_b^l U)) \tag{3}$$

[2]We denote the a_t as uninstantiated argument and x_t as instantiated argument with an AC.

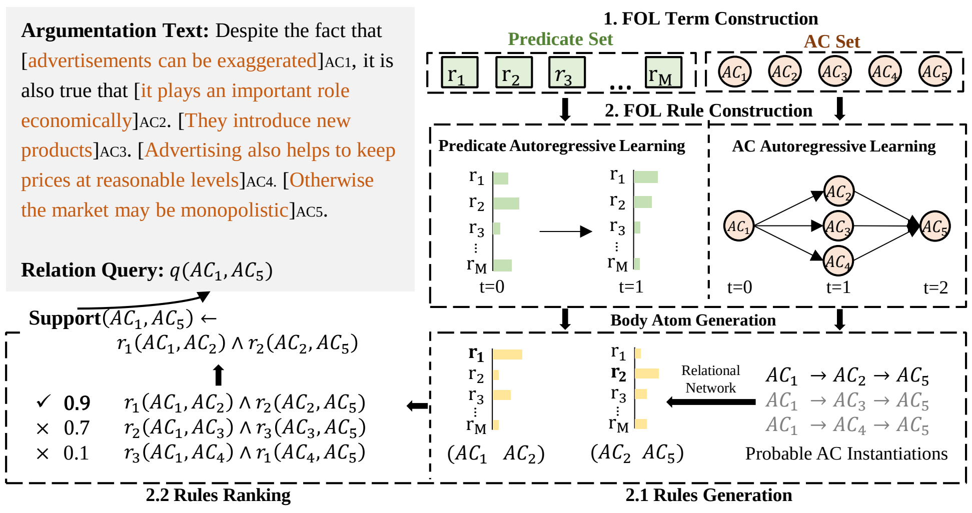

Argumentation Text: Despite the fact that [advertisements can be exaggerated]AC1, it is also true that [it plays an important role economically]AC2. [They introduce new products]AC3. [Advertising also helps to keep prices at reasonable levels]AC4. [Otherwise the market may be monopolistic]AC5.
Relation Query: $q(AC_1, AC_5)$
1. FOL Term Construction
Predicate Set
AC Set
2. FOL Rule Construction
Predicate Autoregressive Learning
AC Autoregressive Learning
t=0
t=1
t=2
Body Atom Generation
Relational Network
$AC_1 \to AC_2 \to AC_5$
$AC_1 \to AC_3 \to AC_5$
$AC_1 \to AC_4 \to AC_5$
Probable AC Instantiations
$(AC_1\ AC_2)$
$(AC_2\ AC_5)$
Support$(AC_1, AC_5) \leftarrow r_1(AC_1, AC_2) \wedge r_2(AC_2, AC_5)$
✓ 0.9 $r_1(AC_1, AC_2) \wedge r_2(AC_2, AC_5)$
× 0.7 $r_2(AC_1, AC_3) \wedge r_3(AC_3, AC_5)$
× 0.1 $r_3(AC_1, AC_4) \wedge r_1(AC_4, AC_5)$
2.2 Rules Ranking
2.1 Rules Generation

Figure 2: The overview of our approach, with a running example for classifying the relation between an AC pair.

$$\mathbf{u}_{t+1}^l = \mathbf{u}_t^l + \mathbf{s}_t^l \cdot (W_b^l U) \tag{4}$$

where $\mathbf{u}_0^l = \mathbf{u}^l$, W_t^l and W_b^l are learnable weight matrices. Intuitively, to learn to define a l-hop rule body, Eq. 3 and 4 sequentially produce predicate r_t at each step $t \in \{1, ..., l\}$ by attending to all the predicates with probability s_t^l.

Body Atom Generation We integrate the selected ACs and predicates to generate the body atoms of a rule body. However, the AC and predicate autoregression learning ignore the correlation between predicates and ACs. This causes always generating the same predicates for different selected AC pairs in each step and has a pretty poor performance in Section 4.5. To address this, we propose a relational network that produces an AC-specific probability distribution $\mathbf{p}(x_t, x_{t+1}) \in \mathbb{R}^M$ over all predicates based on selected AC pairs x_t, x_{t+1}. This distribution denotes the compatibility of selected AC pair (x_t, x_{t+1}) with each predicate $r_t \in \mathcal{R}$ to form an atom $r_t(x_t, x_{t+1})$:

$$\mathbf{p}(x_t, x_{t+1}) = \mathrm{softmax}(W^r \tanh[\mathbf{h}_t; \mathbf{h}_{t+1}]) \tag{5}$$

where W^r is a weight matrix.

We combine the probabilities from predicate autoregression learning with AC-specific probabilities to compute the value [3] $\mu(r_t(x_t, x_{t+1}))$ of an body atom $r_t(x_t, x_{t+1})$ being true formed by the predicate r_t and two ACs (x_t, x_{t+1}):

$$\mu(r_t(x_t, x_{t+1})) = \max(\mathbf{s}_t^l \odot \mathbf{p}(x_t, x_{t+1})) \tag{6}$$

[3]For smooth optimization, we use $\mu(\cdot) \in [0, 1]$ to denote the value of an atom or a rule body, which indicates the probability of the atom or rule body being true.

where $\odot$ denotes element-wise multiplication and r_t is selected as the entry with the highest probability. Note that the predicate autoregression learning captures the sequential dependency among predicates and the relational network models the correlation between the predicate and ACs. By combining the two information, we can construct more robust reasoning rules.

Finally, we can combine the body atoms $r_t(x_t, x_{t+1})(1 \le t \le l)$ into a complete rule body. Since the AC instantiations of bridging arguments are variable, we input each probable AC pair from AC autoregression learning into Eq. 5 and 6 for each body atom, forming multiple probable rule bodies. For example, consider the case in Figure 2 where $l = 2$. Given the possible AC instantiations (x_1, x_2, x_5) for all arguments, Eq. 5 and 6 first generate a body atom $r_1(x_1, x_2)$ based the AC pair (x_1, x_2). Then, the next AC pair (x_2, x_5) is used to generate another predicate r_2, forming the second body atom $r_2(x_2, x_5)$. As a result, the rule body $r_1(x_1, x_2) \wedge r_2(x_2, x_5)$ is constructed.

3.2.2 Rules Ranking

These rule body candidates need to be evaluated and selected to induce the relation query in the head atom. We compute the value of each probable rule body being true. Given a l-hop rule body consisting of body atoms (i.e., $r_t(x_t, x_{t+1})$) and conjunction operators ($\wedge$), computing its value is difficult due to its discrete nature (Qu et al., 2020). Here, we introduce T-Norm logic (Klement et al., 2013) to relax the discrete nature of the rule body evaluation, enabling an end-to-end learning process. The T-Norm logic defines the value of a rule with two atoms and conjunction operator as $T(\mu_1 \wedge \mu_2) = min(\mu_1, \mu_2)$, where $\mu_1, \mu_2 \in [0, 1]$ refer to the values of two atoms. Thus, the value of the l-hop rule body based on the atom values can be computed as $\min(\{\mu(r_t(x_t, x_{t+1}))\}_{t=0}^{l})$.

We rank all probable rule bodies over all possible AC instantiations and use a max operator to select the rule body with the highest value, representing the truth value μ^l of the l-hop rule body.

$$\mu^l = \max_{x \in \mathcal{X}}(\min(\{\mu(x_t, x_{t+1})\}_{t=1}^{l})) \tag{7}$$

where $\mathcal{X} = \{x | x = (x_1, x_2^k, ..., x_l^k, x_{l+1})\}_{k=1}^{K}$ denotes the set for all possible AC instantiations for the l-hop rule body. The selected ACs and predicates form the rule body with the maximum score. In this way, the generated l-hop FOL rule is $q(x_1, x_{l+1}) \leftarrow r_1(x_1, x_2) \wedge \cdots \wedge r_l(x_l, x_{l+1})$, suggesting a reasoning path for deducing the relation between x_1 and x_{l+1}.

The number of reasoning hops l required to derive the query head should be flexibly decided by the model itself. We iterate through the rule generation and ranking process for different l-hop ($1 \le l \le L$) rules and obtain the values $\{\mu^1, \cdots, \mu^L\}$ of different l-hop FOL rules. Finally, the value $\mu(q(x_1, x_{L+1}))$ of the head atom $q(x_1, x_{L+1})$ is computed by selecting the maximum score over the L rule bodies:

$$\begin{aligned} &\mu(q(x_1, x_{L+1})) = \max_{1 \le l \le L}(\exp(\mathbf{s} \log([\mu^1; \cdots; \mu^L]))) \\ &\mathbf{s} = \mathbf{v}_q[\mathbf{u}^1, \ldots, \mathbf{u}^L] \end{aligned} \tag{8}$$

where $\mathbf{s}$ is a selective distribution over the L rule bodies with different hop and $\mathbf{v}_q$ denotes the learnable vector of the relation label q between the AC pair (x_1, x_{L+1}).

Training and Inference The generated FOL rules need to be tested and refined against the ground-truth label, known as learning from entailment, which maximizes the probabilities of the ground-truth label and minimizes those of other labels. We use cross-entropy loss to train the model during the training process:

$$L = -log \frac{\mu(\hat{q}(x_i, x_j))}{\sum_{q \in Y} \mu(q(x_i, x_j))} \tag{9}$$

where $\hat{q}(x_i, x_j)$ denotes the ground-truth label for AC pair (x_i, x_j) in ACTC (ARI) and Y denotes the label set Y_{ACTC} (Y_{ARC}) of ACTC (ARC). The choice of the label set depends on the type of relation query $q(x_i, x_j)$. For example, if x_j denotes the specific AC x_w, then Y_{ACTC} is used; otherwise, Y_{ARC} is applied. During inference, we query all labels in ACTC (ARC) and select the label with the highest truth value of the reasoning rule as the prediction result. Although our methods apply autoregressive learning to produce instantiated ACs and predicates, our use of parallel tensor operations allows it to perform as efficiently as previous methods. The time cost of FOL-AM is detailed in Appendix A.6.

3.3 Adaptiveness for LLM

Inspired by recent advancements in LLMs (Minaee et al., 2024), we extend our FOL-AM framework to leverage LLMs, prompting them to perform FOL-based reasoning for AM. The reasoning process in LLM-enhanced FOL-AM$_{PT}$ [4] closely follows the fine-tuned FOL-AM$_{FT}$ approach, including both *FOL Term Construction* and *FOL Rule Generation*.

FOL Term Construction We construct the fundamental FOL terms, including arguments and predicates, within the FOL rule. ACs are used to instantiate the arguments. Unlike fine-tuned FOL-AM, which employs shared parameterized predicates across all samples, the predicates in the LLM-based approach are dynamically derived for each sample. Specifically, we input the argumentative text W and each AC pair (x_i, x_j) into an LLM and instruct it to generate multiple relation phrases $R_{i,j}$ as potential predicates for the given AC pair.

FOL Rule Generation To generate FOL rules, we first identify relevant ACs and predicates. Given a relation query $q(x_i, x_j)$ for ACTC (ARC) as the head atom, we provide the argumentative text W and the AC set X as input to the LLM and instruct it to select the three most relevant ACs from X for each AC in the head atom. These selected ACs instantiate the bridging arguments in the FOL rules, while the relation phrases between them serve as potential predicates for each AC pair. Next, we input the selected ACs, along with all probable predicates for each AC pair, into the LLM and instruct it to generate several candidate rule bodies for the head atom.

Finally, we perform *FOL rule ranking* by instructing the LLM to identify the most plausible rule body from the generated rule set and use it

[4] We distinguish two implementations: *FOL-AM$_{FT}$* using task-specific fine-tuning and *FOL-AM$_{PT}$* employing LLM prompting. Unless otherwise specified, FOL-AM refers to the *FOL-AM$_{FT}$*.

Model	ACTC	ARI	ARTC	Avg
Joint-ILP	82.6	58.5	-	-
Joint-PN	84.9	60.8	-	-
BERT-Trans	88.4	70.6	-	-
LSTM+dist	85.7	67.8	54.3	69.3
BART	82.7	62.9	53.4	66.3
ST	86.8	69.3	57.1	71.1
PITA	<u>88.3</u>	<u>73.5</u>	<u>59.2</u>	<u>73.7</u>
FOL-AM (our)	**89.3**	**75.7**	**64.0**	**76.3 (+2.6)**

Table 1: Performance comparison on the PE dataset. **Avg** indicates the average value across all metrics. Our improvements over baselines are statistically significant with $p < 0.05$.

Model	ACTC	ARI	ARTC	Avg
SSVM-strict	73.2	26.7	-	-
TSP-PLBA	78.9	34.0	-	-
BERT-Trans	82.5	37.3	-	-
DR-LG	65.3	29.3	15.0	36.5
BART	81.4	32.9	16.7	43.7
ST	82.3	40.2	20.4	47.6
PITA	<u>83.6</u>	<u>44.9</u>	<u>23.8</u>	<u>50.8</u>
FOL-AM (our)	**85.3**	**47.7**	**28.7**	**53.9 (+3.1)**

Table 2: Performance comparison on CDCP dataset. **Avg** indicates the average value across all metrics. Our improvements over baselines are statistically significant with $p < 0.05$.

for label prediction. Specifically, the LLM is directed to select the most relevant rule body and apply logical reasoning within the FOL framework to determine the final label for the AC pair in the head atom for ACTC (ARC). The details of the prompt template are provided in Appendix A.7.

4 Experiments

4.1 Experimental Settings

We evaluate our approach on two AM benchmark datasets: PE (Stab and Gurevych, 2017b) and CDCP (Park and Cardie, 2018). The statistical details of these datasets are in Appendix A.1. For evaluation, we report the macro-averaged F1 score (Macro) for ACTC. Following prior work (Morio et al., 2022b; Sun et al., 2024), we adopt a hierarchical evaluation protocol for ARC, decomposing it into two metrics: ARI, which measures the F1 score for identifying whether an argumentative relation exists between a pair of ACs; and ARTC, which measures the Macro score for correctly classifying the type of relation (e.g., Support and Attack for PE; Reason and Evidence for CDCP) for AC pairs where a relation is present.

For optimization, we use AdaW (Loshchilov and Hutter, 2017) with a learning rate of $3e{-}5$ and weight decay of $1e{-}2$ for both PE and CDCP datasets. The reasoning hops L is set to 2 for both datasets. The number of predicates M is set to 40 for PE and 30 for CDCP, while the number of instantiations K of bridging argument is set to 5 for PE and 4 for CDCP. To reduce randomness, we repeat the experiments three times with different seeds and report the average results. More implementation details are in Appendix A.3. Our code is available at `https://github.com/syiswell/logic-AM`.

4.2 Baselines

We select strong AM models from recent years as baselines, following previous work (Morio et al., 2022b). For the PE dataset, we compare our FOL-AM model with seven baselines: **Joint-ILP** (Stab and Gurevych, 2017b), **Joint-PN** (Potash et al., 2017b), **LSTM+dist** (Kuribayashi et al., 2019b), **BERT-Trans** (Bao et al., 2021a), **BART** (Lewis et al., 2019), **ST** (Morio et al., 2022b), and **PITA** (Sun et al., 2024). For the CDCP dataset, we compare with seven baselines: **DR-LG** (Galassi et al., 2018), **SSVM-strict** (Niculae et al., 2017), **TSP-PLBA** (Morio et al., 2020b), **BERT-Trans** (Bao et al., 2021a), **BART** (Lewis et al., 2019), **ST** (Morio et al., 2022b), and **PITA** (Sun et al., 2024).

Additionally, we apply our prompt-based FOL-AM framework to two large language models (LLMs): GPT-4o-mini (OpenAI, 2024) and Llama-3.1-8B (Dubey et al., 2024). We compare our prompt-based FOL-AM$_{PT}$ with two baselines: (i) *Standard LLMs*, which predict argumentation labels directly without reasoning, and (ii) *Chain-of-Thought (CoT)* (Wei et al., 2022), which generates intermediate explanations before making final predictions. The experiments are conducted under both zero-shot and few-shot settings.

4.3 Overall Performance

The experimental results for both the PE and CDCP datasets are summarized in Table 1 and Table 2, respectively. We observe that FOL-AM consistently achieves superior performance across both datasets. Specifically, on the PE dataset, our FOL-AM model outperforms the state-of-the-art (SOTA) model PITA by 4.8% in ARC and 2.6% in Avg. A similar trend is observed for the CDCP dataset, where FOL-AM surpasses PITA by 4.9% in ARC and 3.1% in Avg.

Additionally, all models exhibit a consistent performance gap between ARI and ARTC scores (e.g.,

Data	LLM	Method	Zero-Shot				Few-Shot			
			ACTC	ARI	ARTC	Avg	ACTC	ARI	ARTC	Avg
PE	LLaMA-3.1-8B	Standard	28.7	49.0	24.9	34.2	35.8	49.5	25.1	36.8
		CoT	36.0	49.3	30.5	38.6	47.1	49.4	30.3	42.3
		FOL-AM$_{PT}$	35.6	50.7	31.2	39.2	44.4	51.0	32.2	42.5
	GPT-4o-mini	Standard	37.4	53.7	31.4	40.8	53.9	54.4	38.6	49.0
		CoT	45.8	51.9	33.3	43.7	56.9	52.3	39.1	49.4
		FOL-AM$_{PT}$	**48.6**	**54.6**	**36.2**	**47.1 (+3.4)**	**58.2**	**56.0**	**41.4**	**51.9 (+2.5)**
CDCP	LLaMA-3.1-8B	Standard	44.9	13.6	7.5	22.0	42.0	13.7	8.9	21.5
		CoT	50.2	15.9	8.2	24.8	50.7	14.5	8.0	24.4
		FOL-AM$_{PT}$	53.5	15.4	9.1	26.0	54.3	14.4	8.4	25.7
	GPT-4o-mini	Standard	50.8	23.9	12.7	29.1	62.7	21.2	12.9	32.2
		CoT	56.0	23.4	13.4	31.0	57.2	23.8	13.3	31.4
		FOL-AM$_{PT}$	**63.1**	**24.8**	**14.3**	**34.1 (+3.1)**	**65.2**	**25.1**	**14.8**	**35.0 (+2.8)**

Table 3: Performance comparison of LLMs on PE and CDCP datasets under zero-shot/few-shot settings, presenting the $F1$ for ARI and Macro for ACTC and ARC. Avg indicates the average value across all metrics.

11.7 points for FOL-AM and 14.3 points for PITA on the PE dataset), underscoring the greater difficulty of fine-grained relation classification (i.e., ARTC) compared to binary relation identification (i.e., ARI). Notably, FOL-AM shows a smaller drop between ARI and ARTC scores than baseline models, suggesting that it more effectively preserves argumentative semantics—likely due to the explicit logical constraints imposed during relation classification. These consistent improvements highlight the effectiveness of integrating FOL rules to explicitly capture reasoning patterns in AM.

4.4 Adaptability Experiment

Table 3 shows the results of LLM-based methods for AM. We observe that CoT-based methods consistently outperform standard methods, emphasizing the crucial role of reasoning ability in AM. Our prompt-based FOL-AM$_{PT}$ methods significantly surpass both baselines (i.e., standard and CoT) across all metrics on PE and CDCP, demonstrating the effectiveness of our framework in enhancing model reasoning capabilities. Moreover, few-shot methods generally outperform zero-shot methods, and our FOL-AM$_{PT}$ methods improve performance in both settings, further underscoring its universality. Notably, our fine-tuned FOL-AM$_{FT}$ model in Table 1 and 2 significantly outperforms all zero-shot and few-shot methods. This improvement stems from its ability to learn logic rules directly from the dataset, thereby enhancing its reasoning capacity. In contrast, zero-shot and few-shot methods rely on the model's inherent reasoning ability and a limited number of examples, leading to lower performance. Given the strong performance of our fine-tuned FOL-AM$_{FT}$ model, we conduct further experiments based on it in the following sections.

Model	PE			CDCP		
	ACTC	ARI	ARTC	ACTC	ARI	ARTC
FOL-AM	**89.3**	**75.7**	**64.0**	**85.3**	**47.7**	**28.7**
w/o PG	85.4	71.4	50.2	73.1	45.4	23.4
w/o PAL	77.3	60.3	41.1	66.2	29.8	14.3
w/o RN	79.7	65.9	43.8	72.6	44.3	22.5
w/o AAL	88.4	74.6	61.1	84.8	47.5	24.3

Table 4: The impact of various components presenting the $F1$ for ARI and Macro for ACTC and ARC.

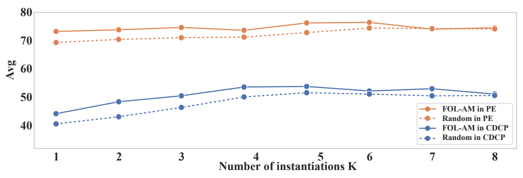
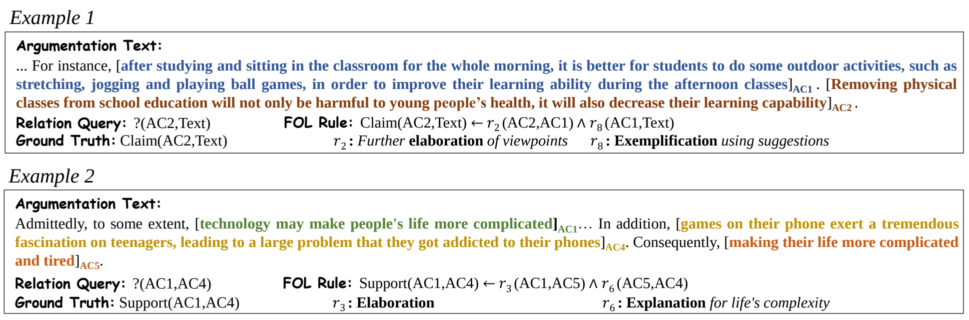


Figure 3: The impact of the number of AC instantiations on PE and CDCP.

4.5 Ablation Study

Table 4 compares our FOL-AM method with several variants where different components are removed: *w/o predicate generation (w/o PG)*, which eliminates predicate generation and instead computes only binary relations to determine whether two ACs are related; *w/o predicate autoregression learning (w/o PAL)*, which disregards the sequential dependencies among predicates; and *w/o relational network (w/o RN)*, which removes the relational network and computes the value of a body atom by simply multiplying the similarity of the AC pair with the predicate probability distribution. *w/o AC autoregression learning (w/o AAL)*, which selects all ACs as bridging arguments. The results show that predicate generation is crucial for reasoning, as its removal causes a significant performance drop. Similarly, removing predicate autoregression learn-



Example 1

Argumentation Text:
... For instance, [**after studying and sitting in the classroom for the whole morning, it is better for students to do some outdoor activities, such as stretching, jogging and playing ball games, in order to improve their learning ability during the afternoon classes**]$_{AC1}$. [**Removing physical classes from school education will not only be harmful to young people's health, it will also decrease their learning capability**]$_{AC2}$.
Relation Query: ?(AC2,Text) **FOL Rule:** Claim(AC2,Text) $\leftarrow r_2$(AC2,AC1) $\wedge\, r_8$(AC1,Text)
Ground Truth: Claim(AC2,Text) r_2 **:** *Further* **elaboration** *of viewpoints* r_8 **: Exemplification** *using suggestions*

Example 2

Argumentation Text:
Admittedly, to some extent, [**technology may make people's life more complicated**]$_{AC1}$… In addition, [**games on their phone exert a tremendous fascination on teenagers, leading to a large problem that they got addicted to their phones**]$_{AC4}$. Consequently, [**making their life more complicated and tired**]$_{AC5}$.
Relation Query: ?(AC1,AC4) **FOL Rule:** Support(AC1,AC4) $\leftarrow r_3$(AC1,AC5) $\wedge\, r_6$(AC5,AC4)
Ground Truth: Support(AC1,AC4) r_3 **: Elaboration** r_6 **: Explanation** *for life's complexity*

Figure 4: Examples of interpretability of our method for ACTC and ARC.

ing harms performance, highlighting the importance of sequential dependencies among predicates. Omitting the relational network also reduces performance by losing the correlation between predicates and ACs. Lastly, the *w/o AAL* variant suffers from degraded performance, confirming precise AC selection is vital for reasoning rule modeling.

4.6 Impact of AC Instantiation

Figure 3 examines the effect of varying the number of instantiations K of bridging argument in AC autoregression learning on the PE and CDCP datasets. The value of K determines the number of probable rule bodies. The results indicate that performance remains relatively stable across both datasets as K varies. However, when $K = 1$, performance is consistently the lowest, reinforcing our assumption that AC autoregression learning struggles to find the global optimum in this scenario. Additionally, when we randomly select K ACs as bridging arguments (denoted as *Random*) rather than employing the AC autoregression learning (AAL) mechanism, the performance of the *Random* variant is consistently worse than that of the model with AAL. This suggests that similarity-based AC selection through AAL enables the model to more effectively capture and utilize logical dependencies among ACs, ultimately leading to improved performance.

4.7 Impact of Reasoning Hops in FOL Rules

Table 5 analyzes the impact of reasoning hops in FOL rules. The *Strict* setting enforces an exact L-hop reasoning approach, while the *Adaptive* setting allows for a maximum of L reasoning hops, selecting the prediction with the highest score adaptively. Notably, the 1-hop setting remains identical in both *Strict* and *Adaptive* configurations.

The results indicate that the *Strict* setting leads to suboptimal performance, whereas the *Adaptive* setting achieves significantly better results. Furthermore, the 1-hop FOL rule performs poorly, suggesting that relation queries in AM benefit more from multi-hop reasoning patterns.

Setting	L	PE			CDCP		
		ACTC	ARI	ARTC	ACTC	ARI	ARTC
Strict	One	73.5	70.1	49.9	59.3	35.8	17.7
	Two	78.6	66.3	44.4	69.5	28.7	15.2
	Three	76.9	63.2	40.3	66.9	25.2	13.9
	Four	74.1	61.8	39.6	59.9	22.4	12.1
Adaptive	Two	**89.3**	**75.7**	**64.0**	**85.3**	**47.7**	**28.7**
	Three	86.8	73.9	60.6	83.4	44.1	23.0
	Four	83.2	69.5	49.2	79.2	39.2	20.3

Table 5: The impact of varying FOL rule's lengths in different FOL settings. The "Strict" setting refers to precise adoption with fixed hop counts, while the "Adaptive" setting allows up to L-hop, with "one hop" indicating no bridging AC.

4.8 Interpretability and Case Study

Figure 4 examines the interpretability of our method through a case study for ACTC and ARC. The results demonstrate that FOL-AM effectively generates FOL rules with selected bridging ACs and predicates to infer the relation query. In the first example, the type of AC2 is determined by leveraging bridging AC1, with r_2 interpreted as *elaboration* and r_8 as *exemplification*. The second case highlights the ability of FOL-AM to interpret multi-hop reasoning in identifying relations between distant AC pairs. Here, r_3 serves as a *summary* connecting two similar viewpoints, while r_6 explains why "life is complicated." Although AC1 and AC4 appear to describe different topics at the surface semantic level, a *Support* relation between them can be inferred through the bridging AC5.

5 Related Works

5.1 Argumentation Mining

Argumentation Mining (AM) is a computational task that aims to automatically identify and extract argument structures from argumentative texts (Lawrence and Reed, 2020). With the advent of deep learning, Laha and Raykar (2016) introduced the first neural model for AM. Later, studies incorporated linguistic features and discourse relations for AM (Kuribayashi et al., 2019b; Chakrabarty et al., 2019). Recent models used advanced architectures like biaffine attention (Morio et al., 2020a, 2022b), transition-based network (Bao et al., 2021a) and Graph Convolutional Networks (Sun et al., 2024). Additionally, some studies explored integrating knowledge bases for logical reasoning. (Moens, 2018) enhanced AM with world knowledge, while (Paul et al., 2020) used unsupervised graph methods for multi-hop reasoning. In contrast, our approach employs FOL rules to model reasoning directly without relying on predefined rules or external knowledge bases.

5.2 FOL Rule

First-order logic (FOL) rules provide a rigorous foundation for formal reasoning, enabling precise knowledge representation and logical deduction (Hughes, 1996). In the deep learning era, several studies explored integrating FOL rules with neural networks for tasks, such as knowledge base inference (Qu et al., 2020), text entailment (Li and Srikumar, 2019), question answering (Wang and Pan, 2022), and event argument extraction (Liu et al., 2023). With the rise of LLMs, integrating FOL reasoning into LLMs has gained attention (Ye et al., 2023; Pan et al., 2023; Olausson et al., 2023; Xu et al., 2024). However, these approaches are not directly applicable to AM due to differences in task structure and the reasoning required. AM involves identifying and classifying complex arguments and relations in textual contexts, which existing methods may not capture effectively. Our work is inspired by knowledge base inference but addresses the unique challenges of AM. To the best of our knowledge, FOL-AM is the first attempt to unify multiple AM tasks into a relation query framework while incorporating FOL reasoning.

6 Conclusion

In conclusion, we introduce the FOL-AM framework, which enhances argumentative structure extraction by explicitly capturing logical reasoning paths through FOL rules. The framework, implemented with both fine-tuning and LLM prompt-based methods, achieves superiority performance and offers enhanced explainability on AM benchmarks, demonstrating its effectiveness for AM.

Limitation

To outline potential future research directions for AM, we conduct a detailed error analysis on 100 cases where FOL-AM made incorrect predictions. Our findings highlight two main challenges.

First, FOL-AM struggles with classifying instances that rely on finer-grained bridging units for ACTC. For example, expressions such as "Thus, it is apparent that..." and "I strongly believe that..." typically indicate a *Claim*. However, since ACs are defined as FOL terms, these expressions may not constitute an entire AC or may only form a small segment within a longer AC. As a result, the model tends to overlook these fine-grained features during classification. To address this limitation, we suggest that future research explore multi-hop reasoning in AM by incorporating finer-grained bridging units, such as elementary discourse units.

Second, despite our adaptive multi-hop reasoning mechanism, FOL-AM tends to overconstruct long reasoning paths for irrelevant AC pairs. This issue becomes particularly pronounced as the rule length increases, likely due to inaccuracies or error propagation in the intermediate steps of multi-hop reasoning. To mitigate this problem, we propose designing a loss function that penalizes excessive rule length, encouraging the model to generate more effective and concise FOL reasoning chains. Such an approach would help capture AC relationships more accurately while reducing the risk of error propagation.

By addressing these challenges, future work can further enhance the interpretability and robustness of FOL-based reasoning in AM.

Acknowledgements

We thank the anonymous reviewers for their valuable suggestions to improve the quality of this work. This work was partially supported by the National Key Research and Development Program of China (2022YFF0902100), the National Natural Science Foundation of China 62176076, Natural Science Foundation of Guang Dong (2023A1515012922, 2024A1515030166), the Shenzhen Foundational Research Funding (KQTD20190929172835662, JCYJ20220818102415032), the Major Key Project of (PCL2021A06, PCL2023A09), Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies 2022B1212010005 and CIPSC-SMP-ZHIPU Large Model Cross-Disciplinary Fund ZPCG20241119405.

References

Jianzhu Bao, 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 (Volume 1: Long Papers)*, pages 6354–6364.

Jianzhu Bao, Chuang Fan, Jipeng Wu, Yixue Dang, Jiachen Du, and Ruifeng Xu. 2021b. 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, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6354–6364. Association for Computational Linguistics.

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

Tuhin Chakrabarty, Christopher Hidey, Smaranda Muresan, Kathy McKeown, and Alyssa Hwang. 2019. AMPERSAND: argument mining for persuasive online discussions. 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, Hong Kong, China, November 3-7, 2019*, pages 2933–2943. Association for Computational Linguistics.

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

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

Xinyu Hua, 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, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2661–2672. Association for Computational Linguistics.

GE Hughes. 1996. *A New Introduction to Modal Logic*. Routledge.

Khalid Al Khatib, Lukas Trautner, Henning Wachsmuth, Yufang Hou, and Benno Stein. 2021. Employing argumentation knowledge graphs for neural argument generation. 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, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4744–4754. Association for Computational Linguistics.

Erich Peter Klement, Radko Mesiar, and Endre Pap. 2013. *Triangular norms*, volume 8. Springer Science & Business Media.

Tatsuki Kuribayashi, Hiroki Ouchi, Naoya Inoue, Paul Reisert, Toshinori Miyoshi, Jun Suzuki, and Kentaro Inui. 2019a. An empirical study of span representations in argumentation structure parsing. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4691–4698. Association for Computational Linguistics.

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

Anirban Laha and Vikas C. Raykar. 2016. An empirical evaluation of various deep learning architectures for bi-sequence classification tasks. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 2762–2773. ACL.

John Lawrence and Chris Reed. 2019. Argument mining: A survey. *Comput. Linguistics*, 45(4):765–818.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Preprint*, arXiv:1910.13461.

Tao Li and Vivek Srikumar. 2019. Augmenting neural networks with first-order logic. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 292–302.

Jian Liu, Chen Liang, Jinan Xu, Haoyan Liu, and Zhe Zhao. 2023. Document-level event argument extraction with a chain reasoning paradigm. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9570–9583.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Marie-Francine Moens. 2018. Argumentation mining: How can a machine acquire common sense and world knowledge? *Argument Comput.*, 9(1):1–14.

Gaku Morio, Hiroaki Ozaki, Terufumi Morishita, Yuta Koreeda, and Kohsuke Yanai. 2020a. 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, Online, July 5-10, 2020*, pages 3259–3266. Association for Computational Linguistics.

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

Gaku Morio, Hiroaki Ozaki, Terufumi Morishita, and Kohsuke Yanai. 2022a. End-to-end argument mining with cross-corpora multi-task learning. *Trans. Assoc. Comput. Linguistics*, 10:639–658.

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

Vlad Niculae, Joonsuk Park, and Claire Cardie. 2017. Argument mining with structured svms and rnns. *arXiv preprint arXiv:1704.06869*.

Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. 2023. Linc: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5153–5176.

OpenAI. 2024. Chatgpt, prompt. Accessed: 2024-10-28.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824.

Joonsuk Park 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)*.

Debjit Paul, Juri Opitz, Maria Becker, Jonathan Kobbe, Graeme Hirst, and Anette Frank. 2020. Argumentative relation classification with background knowledge. In *Computational Models of Argument - Proceedings of COMMA 2020, Perugia, Italy, September 4-11, 2020*, volume 326 of *Frontiers in Artificial Intelligence and Applications*, pages 319–330. IOS Press.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2017a. 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, Copenhagen, Denmark, September 9-11, 2017*, pages 1364–1373. Association for Computational Linguistics.

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

Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. 2020. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. *arXiv preprint arXiv:2010.04029*.

Robin Schaefer and Manfred Stede. 2021. Argument mining on twitter: A survey. *it Inf. Technol.*, 63(1):45–58.

Noam Slonim, Yonatan Bilu, Carlos Alzate, Roy Bar-Haim, Ben Bogin, Francesca Bonin, Leshem Choshen, Edo Cohen-Karlik, Lena Dankin, Lilach Edelstein, Liat Ein-Dor, Roni Friedman-Melamed, Assaf Gavron, Ariel Gera, Martin Gleize, Shai Gretz, Dan Gutfreund, Alon Halfon, Daniel Hershcovich, Ron Hoory, Yufang Hou, Shay Hummel, Michal Jacovi, Charles Jochim, Yoav Kantor, Yoav Katz, David Konopnicki, Zvi Kons, Lili Kotlerman, Dalia Krieger, Dan Lahav, Tamar Lavee, Ran Levy, Naftali Liberman, Yosi Mass, Amir Menczel, Shachar Mirkin, Guy Moshkowich, Shila Ofek-Koifman, Matan Orbach, Ella Rabinovich, Ruty Rinott, Slava Shechtman, Dafna Sheinwald, Eyal Shnarch, Ilya Shnayderman, Aya Soffer, Artem Spector, Benjamin Sznajder, Assaf Toledo, Orith Toledo-Ronen, Elad Venezian, and Ranit Aharonov. 2021. An autonomous debating system. *Nat.*, 591(7850):379–384.

Christian Stab and Iryna Gurevych. 2014. Annotating argument components and relations in persuasive essays. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers*, pages 1501–1510.

Christian Stab and Iryna Gurevych. 2017a. Parsing argumentation structures in persuasive essays. *Comput. Linguistics*, 43(3):619–659.

Christian Stab and Iryna Gurevych. 2017b. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

Yang Sun, Muyi Wang, Jianzhu Bao, Bin Liang, Xiaoyan Zhao, Caihua Yang, Min Yang, and Ruifeng Xu. 2024. PITA: prompting task interaction for argumentation mining. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 5036–5049. Association for Computational Linguistics.

Alexandros Vassiliades, Nick Bassiliades, and Theodore Patkos. 2021. Argumentation and explainable artificial intelligence: a survey. *Knowl. Eng. Rev.*, 36:e5.

Eva Maria Vecchi, Neele Falk, Iman Jundi, and Gabriella Lapesa. 2021. Towards argument mining for social good: A survey. 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, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 1338–1352. Association for Computational Linguistics.

Vern R. Walker, Dina Foerster, Julia Monica Ponce, and Matthew Rosen. 2018. Evidence types, credibility factors, and patterns or soft rules for weighing conflicting evidence: Argument mining in the context of legal rules governing evidence assessment. In *Proceedings of the 5th Workshop on Argument Mining, ArgMining@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 68–78. Association for Computational Linguistics.

Thiemo Wambsganss, Christina Niklaus, Matthias Cetto, Matthias Söllner, Siegfried Handschuh, and Jan Marco Leimeister. 2020. AL: an adaptive learning support system for argumentation skills. In *CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020*, pages 1–14. ACM.

Thiemo Wambsganss and Roman Rietsche. 2019. Towards designing an adaptive argumentation learning tool. In *Proceedings of the 40th International Conference on Information Systems, ICIS 2019, Munich, Germany, December 15-18, 2019*. Association for Information Systems.

Wenya Wang and Sinno Pan. 2022. Deep inductive logic reasoning for multi-hop reading comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4999–5009.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. 2024. Faithful logical reasoning via symbolic chain-of-thought. *arXiv preprint arXiv:2405.18357*.

Jiacheng Ye, Chengzu Li, Lingpeng Kong, and Tao Yu. 2023. Generating data for symbolic language with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8418–8443.

A Appendix

Type	PE	CDCP
Paragraphs	1833	731
Train	1464	581
Test	369	150
Components	6089	4779
Relations	3832	1353
Components Per Sample	3.5	6.5

Table 6: The statistics of the PE and CDCP datasets.

A.1 Data statistics

We evaluate our approach on two popular benchmarks. Table 6 provides detailed dataset statistics.

- **PE** (Stab and Gurevych, 2017b): Comprising 420 essays totaling 1,833 paragraphs, this corpus categorizes argumentation components (ACs) into three classes (*MajorClaim*, *Claim*, *Premise*) and argumentative relations (ARs) into two types (*Support*, *Attack*). The dataset's structural constraint allows each AC to possess a maximum of one outgoing AR, forming directed tree/forest graphs. We maintain consistent data partitioning with prior works (Kuribayashi et al., 2019a; Bao et al., 2021b): 1,464 ACs for training (10% as validation) and 369 ACs for testing.

- **CDCP** (Park and Cardie, 2018): This collection contains 731 paragraphs with finer-grained AC categorization (*Value*, *Policy*, *Testimony*, *Fact*, *Reference*) and AR types (*Reason*, *Evidence*). Unlike PE's tree structure, CDCP permits multiple AR connections from a single AC, enabling non-tree graph configurations. Following recent implementations (Bao et al., 2021a; Morio et al., 2022b), we adopt the standard split: 581 ACs for training (10% as validation) and 150 ACs for testing.

A.2 Baselines

PE Dataset: Our experimental evaluation compares against seven competitive baselines following previous works (Morio et al., 2022b):

- **Joint-ILP** (Stab and Gurevych, 2017b): Utilizes Integer Linear Programming for simultaneous optimization of argumentation component type classification (ACTC) and argumentative relation identification (ARI).
- **Joint-PN** (Potash et al., 2017b): Implements a Pointer Network architecture with attention mechanisms for joint ACTC-ARI learning.
- **BERT-Trans** (Bao et al., 2021a): Neural transition-based approach employing action sequence generation for argument analysis.
- **LSTM+dist** (Kuribayashi et al., 2019b): Pioneers LSTM-minus span representations combined with ELMo embeddings for argument mining.
- **BART** (Lewis et al., 2019): Sequence-to-sequence formulation using BART with target structures.
- **ST** (Morio et al., 2022b): Longformer-based architecture with biaffine scoring for structural prediction.
- **PITA** (Sun et al., 2024): A generation framework with dynamic prompt templates and heterogeneous graph modeling is proposed to explicitly coordinate the interaction of different subtasks.

CDCP Dataset: We evaluate against seven competitive baselines:

- **SSVM-strict** (Niculae et al., 2017): Structured SVM variant with constrained factor graphs for dual-task learning.
- **TSP-PLBA** (Morio et al., 2020b): Features task-specific parameterization and proposition-level biaffine attention.
- **BERT-Trans** (Bao et al., 2021a): Neural transition-based approach employing action sequence generation for argument analysis.
- **DR-LG** (Galassi et al., 2018): Residual networks with link-guided training for joint ACTC, ARI, and ARTC.
- **BART** (Lewis et al., 2019): Sequence-to-sequence formulation using BART with target structures.
- **ST** (Morio et al., 2022b): Longformer-based architecture with biaffine scoring for structural prediction.
- **PITA** (Sun et al., 2024): A generation framework with dynamic prompt templates and heterogeneous graph modeling is proposed to explicitly coordinate the interaction of different subtasks.

A.3 More Implementation Details

We implement the proposed framework using the Transformers library (Wolf et al., 2020) in PyTorch and conduct experiments on a GPU. The model is optimized with AdaW (Loshchilov and Hutter, 2017), utilizing a learning rate of $3e{-}5$ and a weight decay of $1e{-}2$ for both the PE and CDCP datasets. The batch size is set to 4 for both datasets. For both datasets, we adopt dropout with a dropout rate of 0.1 to avoid overfitting. We set the reasoning hop of L to 2 for both datasets. We set the number of predicates M to 40 for PE and 30 for CDCP, while the number of instantiations K for bridging argument is set to 5 and 4, respectively. To mitigate randomness, we repeat our approach three times with different random seeds and report the average results.

A.4 Overall Performance Supplement

The experimental results for both the PE and CDCP datasets are summarized in Table 1 and Table 2, respectively. We observe that FOL-AM consistently achieves superior performance across both datasets. Specifically, on the PE dataset, our FOL-AM model outperforms the state-of-the-art (SOTA) model PITA by 4.8% in ARC and 2.6% in Avg. A similar trend is observed for the CDCP dataset, where FOL-AM surpasses PITA by 4.9% in ARC and 3.1% in Avg. These consistent improvements highlight the effectiveness of integrating FOL rules to explicitly capture reasoning patterns in AM.

Further analysis reveals three key trends: (i) Traditional feature-engineered models (Joint-ILP, SSVM-strict) exhibit limited competitiveness due to constraints imposed by manual feature engineering; (ii) Pretrained language model-based approaches (BERT-Trans, ST, and PITA) significantly outperform sequential architectures (LSTM+dist,

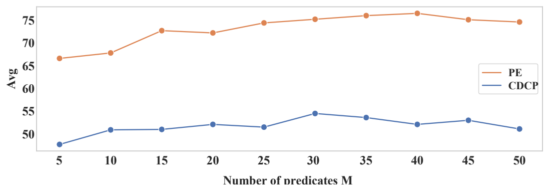


Figure 5: The impact of the number of predicates. Avg indicates the average value across all metrics.

TSP-PLBA), suggesting that PLMs' ability to leverage extensive linguistic knowledge benefits argument structure modeling; (iii) Our logic-enhanced framework consistently surpasses all baselines. This improvement aligns with our design objective, demonstrating that explicit integration of logical rules aids in disentangling complex argumentative dependencies that pure representation-learning-based methods may overlook.

A.5 Impact of different number of predicates

Figure 5 analyzes the effect of the number of predicates on the performance of the PE and CDCP datasets. The results indicate that performance improves initially but declines as the number of predicates increases, with the optimal performance observed at 40 predicates for PE and 30 for CDCP. This finding suggests that a moderate number of predicates achieves a balance between capturing multi-hop reasoning paths and mitigating overfitting. An excessive number of predicates may introduce redundant rules, leading to conflicts with essential reasoning steps and diminishing overall model performance.

Data	**Model**	**TT (min)**	**IT (sec)**
PE	BART	1.17	15.00
	ST	1.23	4.08
	PITA	1.27	2.54
	FOL-AM	1.42	9.16
CDCP	BART	0.63	18.63
	ST	0.81	1.75
	PITA	0.73	1.26
	FOL-AM	0.68	4.83

Table 7: Computational cost in terms of Training Time (TT) per epoch (minutes) and Inference Time (IT) in the test set (second).

A.6 Computational Cost

We evaluate the computational cost of baseline methods and our FOL-AM model during both training and inference. For a fair comparison, all models use a batch size of 4 for both training and inference. Table 7 presents the training and inference times on the PE and CDCP datasets. FOL-AM demonstrates competitive efficiency compared to baselines in training and inference. For instance, FOL-AM increases inference time by 6.62 seconds and 3.57 seconds for all samples in PE and CDCP, respectively, compared to PITA. These datasets contain 369 and 150 samples, respectively. The average increase in time per sample is 0.018 seconds for PE and 0.023 seconds for CDCP. Given these small overheads, the computational cost remains acceptable for practical applications.

A.7 Prompt Templates for LLM

The prompt templates of LLM-based FOL-AM are shown in Figure 6, 7, 8 and 9.

Prompt 1: FOL Term Construction.

```
Given an argumentative text, a pair of argumentative components within the paragraph, please identify the predicates between the argumentative component pair based on the definition of labels. Several words or phrases can represent predicates.

<Definiton of Each Label>
{label definition}
</Definiton of Each Label >

<Examples>
{examples}
</Examples>

Based on the above definitions, you can extract according to the following steps: Observe the definition of each label in turn, and then identify the predicates in the argumentative component pair that have commonalities with these definitions.

Use the following JSON format to output the extracted properties:
{
   "The identified predicates": [ no more than 10 predicates ]
}

< Argumentative Text >
{argumentative text}
</Argumentative Text >

<The Argumentative Component Pair>
{argumentative component pair}
</The Argumentative Component Pair>

Let's think step by step.
```

Figure 6: The format of natural language prompts for FOL term construction.

Prompt 2: Generation of Bridging Arguments.

```
Given an argumentative text, all argumentative components within the paragraph, for a specific argumentative component, please identify the three other argumentative components that are most relevant to it within that paragraph.

<Examples>
{examples}
</Examples>

Use <Answer> and </Answer> to enclose your answer, now please identify the three relevant argumentative components:

<Argumentative Text >
{argumentative text}
</Argumentative Text>

<All Argumentation Components>
{all argumentation components}
</All Argumentation Components>

<The Specific Argumentation Component>
{argumentation component}
</The Specific Argumentation Component>

Let's think step by step.
```

Figure 7: The format of natural language prompts for the generation of bridging arguments.

Prompt 3: FOL Rule Generation.

```
Given an argumentative text, a pair of argumentative components within the paragraph, and a relation graph between the argumentative components, please generate several FOL reasoning chains based on the relation graph.

<Examples>
{examples}
</Examples>


Use the following JSON format to output the inference chains:
{
   “Chain1”: “…”,
   “Chain2": “…”,
   …
}

<Argumentative Text >
{argumentative text}
</Argumentative Text>

<All Argumentation Components>
{all argumentation components}
</All Argumentation Components>

<The Specific Argumentation Component>
{argumentation component}
</The Specific Argumentation Component>

< Relation Graph>
…
</Relation Graph>

Let's think step by step.
```

Figure 8: The format of natural language prompts for FOL rule generation.

Prompt 4: FOL Rule Ranking.

```
Given a argumentation text, a pair of argumentative components within the paragraph, and several FOL reasoning chains, please choose the correct reasoning chain to determine the relation label of the argumentative component pair.

<Definiton of Each Label>
{label definition}
</Definiton of Each Label >

<Examples>
{examples}
</Examples>

Using <Answer> and </Answer> to enclose your answer, now please determine the relation label of the argumentative component pair:

<Argumentative Text >
{argumentative text}
</Argumentative Text>

<All Argumentation Components>
{all argumentation components}
</All Argumentation Components>

<The Specific Argumentation Component>
{argumentation component}
</The Specific Argumentation Component>

< Reasoning Chains>
…
</ Reasoning Chains >

Let's think step by step.
```

Figure 9: The format of natural language prompts for FOL rule ranking.