Argument Structure Prediction in Online Conversations: A Comparative Study of Modeling Paradigms and Task Architectures

Siddharth Bhargava[1,2], Sara Tonelli[1], Patricia Martín-Rodilla[3] and Javier Parapar[2]

[1]*Fondazione Bruno Kessler, Povo 38123, Trento Italy*
[2]*Universidade da Coruña, 15001 A Coruña Spain*
[3]*IEGPS-CSIC, Spanish National Research Council, 15704 Santiago de Compostela Spain*

Abstract

Argument structure prediction (ASP) constructs complete argument structures from discourse by identifying argumentative units and their relations. While recent work has explored diverse approaches—including unified neural models, multi-step pipelines, and prompt-based large language models (LLMs)—their relative trade-offs remain under-explored, particularly in dialogical settings.

We present a systematic evaluation of ASP under strict schema constraints, comparing supervised fine-tuning and prompt-based LLMs across single- and multi-step task architectures, generating complete argument structures from dialogical input end-to-end. We benchmark them on three diverse dialogical corpora adapted from Inference Anchoring Theory into bipolar argument structures. Under a shared evaluation framework, we assess predictive performance, cross-domain generalization, schema compliance, and computational efficiency. Our results show that ASP remains a challenging task, with identifying argumentative relations emerging as the primary bottleneck, largely due to the implicit and context-dependent nature of dialogical argumentation. To facilitate future research, we release our data processing pipeline and end-to-end modeling framework for computational ASP on dialogical corpora.

Keywords

Argument structure prediction, Dialogical analysis, Large language models, Argument mining.

1. Introduction

Modeling argumentative interactions in natural discourse requires capturing how arguments are proposed, challenged, and acknowledged. These interactions are formalized as *argument structures*, i.e., graph-based representations that connect argumentative units through supporting and/or attacking relations. Automatically predicting such structures from raw text is a central task in Argument Mining (AM) [1, 2], commonly referred to as *end-to-end Argument Mining* [3] or *argument structure prediction* (henceforth referred to as ASP in this study) [4] in the literature. Argument structures find application across domains including law, medicine, politics, and scientific writing (e.g., [4, 5, 6, 7]).

Designing and developing ASP requires addressing two key challenges. First, ASP is *theory-driven* requiring a formal representation, grounded in argumentation theory and discourse characteristics. Second, it is *data-driven*, relying on large-scale annotated corpora and supervised learning to automate structured predictions from real-world data. Together, this makes development of ASP systems both methodologically and computationally demanding. These challenges are further amplified in dialogical settings [8, 9], where noisy interactions, implicit argumentative content, and complex discourse dynamics make both formal modeling and automated prediction substantially more difficult, limiting the applicability of existing approaches.

To address these challenges, existing ASP approaches have modeled the task either as a *single-step* multi-task learning problem [2, 10] or as a modular *multi-step* pipeline [11]. More recently, large language

CMNA'26: 26th International Workshop on Computational Models of Natural Argument
sbhargava@fbk.eu (S. Bhargava); satonelli@fbk.eu (S. Tonelli); p.m.rodilla@iegps.csic.es (P. Martín-Rodilla); javier.parapar@udc.es (J. Parapar)
0000-0002-2682-8557 (S. Bhargava); 0000-0001-8010-6689 (S. Tonelli); 0000-0002-1540-883X (P. Martín-Rodilla); 0000-0002-5997-8252 (J. Parapar)


models (LLMs) have been adapted for ASP through prompting or supervised fine-tuning [12, 13, 14]. These approaches differ in their modeling assumptions, training data, and evaluation, making systematic comparisons difficult.

To systematically compare these approaches, we evaluate ASP in dialogical settings across two widely used modeling paradigms: **supervised fine-tuning** and **prompting with self-refinement**. Within each paradigm, we consider two task architectures: as a *single-step* formulation and as a *multi-step* ASP pipeline that first identifies argument units and then classifies the argumentative relations between them. All configurations adhere to a unified argument schema, ensuring consistent inputs and directly comparable outputs. Our proposed experimental design enables us to qualitatively study the trade-offs between modeling paradigms and task architectures, and assess their feasibility for large-scale dialogical argumentation.

We benchmark these approaches on **three** dialogical corpora from AIFdb [15], annotated using Inference Anchoring Theory (IAT) [16] which links dialogical events such as “arguing", “questioning", and “agreeing” to argumentative relations—namely inferential, conflicting, or rephrasing—between pairs of propositions identified in the text. To faciliate the computational modeling of the argument structure, we systematically convert the complex IAT annotations into bipolar argument structures, of argument units linked by either support or attack relations. Our three selected corpora vary in their interaction mode (offline vs. online), topic (political vs. non-political), and dialogue goals (collaborative vs. persuasion), enabling cross-domain evaluation.

The main contributions of this work are as follows:

- We introduce a systematic adaptation of IAT-based dialogical corpora into bipolar argument structures, facilitating benchmarking and computational modeling;
- We develop an open-source, end-to-end pipeline for ASP across multiple task architectures (single-step and multi-step) and modeling paradigms (supervised fine-tuning and prompt-based generative approaches);
- We propose a comprehensive evaluation framework for analyzing performance, generalization, schema compliance, and computational efficiency of argument structures across modeling configurations under a shared schema.

The data and system code have been released for public use (see Appendix A).

2. Related Work

ASP has been studied under different **task architectures**, primarily distinguished by their level of modularization. *Multi-step (modular) approaches* decompose ASP into sequential subtasks such as argument unit identification, argument classification, and relation prediction [11, 17]. Relation prediction may be further divided into relation identification and relation type classification [17, 18]. Each subtask undergoes its own optimization and training before integration.

In contrast, *single-step (joint) approaches* model ASP as a unified structured prediction problem, i.e., jointly modeling argument units and relations within a single optimization and training framework. These methods leverage shared representations across subtasks, including entity-relation formulations [2] and parsing-based or pointer architectures with dialogical extensions [10, 19]. While they reduce error propagation and enable multi-task learning, ensuring structural validity remains challenging. Overall, multi-step approaches favor interpretability and task specialization, whereas single-step models provide integrated learning and resource efficiency at the cost of error propagation. However, their direct comparisons remain limited, especially in shared constraints.

From a **computational modeling** perspective, ASP has seen paradigm shifts from feature-based methods to training neural models and, more recently, adapting LLMs for argument mining [20]. Early feature-engineered approaches [17, 21] offered interpretability but required substantial manual effort and lacked generalization. Neural models improved performance by learning contextual

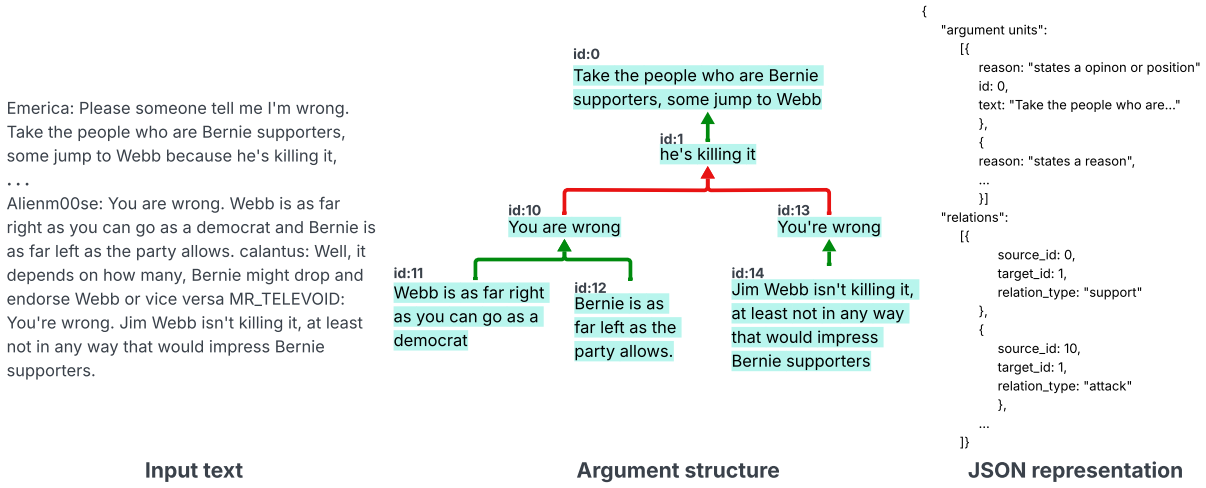


Figure 1: Illustrative example of a conversation excerpt [id:10055] from the US2016reddit corpus (left), the bipolar argument structure representation, annotated by humans (middle)—where green edges denote *support* relations and red edges denote *attack* relations—and the JSON output representation (right) indicating how the argument structure has been stored.

representations [11, 18], though they depend on large annotated datasets and often struggle with out-of-distribution robustness.

Recent LLM-based approaches leverage pretrained knowledge and strong contextual reasoning to generate argument structures [12, 13, 14]. However, they may lack grounding in argumentation theory, offering limited interpretability, and may rely on superficial statistical patterns observed in data rather than robust reasoning [22].

Prior work typically studies the modeling paradigms and task architectures as individual design choices [11, 20]. We address this gap by systematically comparing fine-tuning and prompt-based approaches across both single- and multi-step architectures, evaluating their effect on performance, cross domain generalization, efficiency, and structural validity in diverse dialogical settings.

3. Data

Argument structure prediction in dialogical settings remains a challenging task due to limited annotated resources available to model and train models. To mitigate this, we use AIFdb[1], a large repository of dialogical AM corpora annotated by different AM research teams, and represented in the Argument Interchange Format (AIF) [23]. We adapt the original IAT-annotations to form simplified argument structures that contain support and attack relations (i.e., bipolar nature) between argument units extracted from the raw conversations. We refer to this representation as *bipolar argument structures*, based on bipolar argumentation framework [24], where each structure contains a set of arguments, a set of support relations, and a set of attack relations. In this study, we focus only on IAT-annotated corpora that can be adapted to a bipolar argument structure via a common data processing framework (see Appendix A).

From the IAT annotations, we identify all *locution* nodes (speaker utterances taken verbatim from text) linked by **at least one** *inferential* or *conflict* relation nodes. All locution nodes satisfying this condition form the **argument units**. The resulting inferential and conflict relations are mapped to **support** and **attack** respectively, yielding a bipolar argument structure. Figure 1 illustrates an example conversation with its full representation, linking the raw conversation (left), the resulting argument structure from our adaptation (middle), and the corresponding JSON output (right). All conversations have been stored in this format.

[1] https://corpora.aifdb.org/

Corpus	#Conv. (orig.)	AvgLen	#Units	#Support(%)	#Attack(%)	#LongR(%)
QT30 (train)	669 (1478)	299.2	6451	3580 (85.9)	565 (14.1)	277 (6.7)
QT30 (test)	168 (1478)	304.3	1559	871 (87.1)	141 (12.9)	76 (7.8)
US2016reddit	152 (264)	221.1	1666	895 (71.3)	361 (28.7)	315 (25.6)
RIP1	79 (204)	406.3	765	386 (83.5)	76 (16.5)	113 (25)

Table 1
Corpus statistics after cleaning and filtering. Values in parentheses in the #Conv. column denote the original number of conversations in dataset before filtering. Abbreviations: AvgLen = average conversation length (in tokens); LongR= long-distance relations where source and target are at least 3 sentences apart.

Argument Units. We report in Figure 1 an example of an argument structure where each unit is a span of text expressing a coherent argumentative intent, as based on the IAT framework, extracted verbatim from the conversation. All units include a unique identifier (`id`) and text (`text`) field; Additionally, a `reason` field is included that briefly describes the role of the unit. Identifiers must be unique, sequential, and chronologically ordered (starting from 0). Units must contain at least three words or form a semantically coherent span.

Argument Relations. Relations are directed links between units labeled as *support* or *attack* (Figure 1). Each relation must contain a source unit ID, a target unit ID, and a relation type field. Additionally we maintain that the source–target pairs are chronologically ordered i.e., source unit must occur after target unit. This is compatible with the IAT framework.

Our data pipeline processes IAT corpora (in AIF format) to produce the bipolar argument structures. We have optional support for adding the third *rephrasing* relation type, preserving speaker and dialogical act information. In this study we do not explore rephrasing relations when modeling our argument structures, and leave the investigation for future work. For more details, refer to our data processing pipeline repository (see Appendix A).

For this study, we select three corpora based on size and discourse characteristics: **QT30** (BBC Question Time debates) [8], **US2016reddit** (Reddit reactions to the 2016 U.S. presidential debates) [25], and **RIP1** (collaborative mystery game conversations) [9]. To support fine-tuning and in-context learning, the largest corpus (QT30) is split into training and test sets, adopting a 80:20 split with random state 42 and stratified by attack-relation frequency to preserve class distribution. Since we perform a cross-validation strategy for fine-tuning our models, we do not define an explicit validation set, as explained in Experimental Setup 5.

To maintain data consistency and ensure some argumentative activity, we remove instances with fewer than three units or two relations and limit conversations to 1,500 tokens due to computational constraints. Table 1 presents the number of instances retained following these constraints. Additional preprocessing of the discussion text and argument structure is done to remove non-ASCII characters, markup, duplicates, nested units, and invalid self-relations.

QT30 and US2016reddit contain persuasive political discourse, whereas RIP1 captures collaborative, non-political interaction. As shown in Table 1, US2016reddit contains shorter units and a higher proportion of attacks (approximately 2.5 per conversation) compared to the offline corpora. Overall, attack relations remain relatively sparse across datasets (approximately 1.5 attacks versus 6 supports per conversation). Both US2016reddit and RIP1 exhibit a higher proportion of long-distance relations, while QT30 remains largely local. Although all corpora follow the IAT framework, minor variations arise from differences in annotation practices across research groups. All datasets have been used in accordance with their original licenses and ethical guidelines.

Paradigm	Model	Arch.	Code (MC)
Fine-tuning	BiLSTM-ER	ss	FT
	XLM-R-Long + XLM-R-Large	ms	FT
Prompting	Qwen2.5-14B-Instruct	ss / ms	Q0, Q3, Q5
	DeepSeek-R1-Distill-Qwen-14B	ss / ms	D0, D3, D5
	Gemma3-12B-it	ss / ms	G0, G3, G5

Table 2
Models and configurations evaluated. Paradigms include supervised fine-tuning and prompt-based inference. Task Architectures: single-step (ss) and multi-step (ms). Model codes (MC) denote the model and number of context examples (e.g., Q3 = Qwen with three examples).

4. Models

We investigate two modeling paradigms for ASP: **fine-tuning** of deep learning models and **prompting** large language models. Both are evaluated under two task architectures: a **single-step** formulation, where we adopt a multi-task learning approach and directly produce the argument structure, and a **multi-step** formulation, where the tasks are executed sequentially with intermediate, schema-validated outputs. Table 2 summarizes the models used in this study and introduces the model codes used to refer to the respective configurations in this paper.

4.1. Supervised fine-tuning

For the **single-step** fine-tuning, we adopt the framework proposed in [2], which employs a biLSTM-ER (Entity–Relation) model to perform joint token-level prediction for both unit and relation. The model simultaneously predicts (i) argumentative spans via BIO tagging, (ii) relation types (support or attack), and (iii) relational distance estimates (ranging from -8 to $+8$ sentences) indicating the relative position of the target unit.

For the **multi-step** fine-tuning, we train two models independently for argument unit extraction (AUE) and relation type classification (RTC). The AUE component performs BIO-tagged span-identification using XLM-RoBERTa-Longformer [26] to identify argumentative spans within conversation excerpts with max length 1500. The RTC component performs sequence-level classification using XLM-RoBERTa-large [27] that takes as input a pair of predicted argument units, paired chronologically (i.e., source is always after target) and provided with a limited context window of 150, and determines the relation type from the set $\{$*support*, *attack*, *no-relation*$\}$. To improve discrimination between relational and non-relational pairs, RTC training data is augmented with a proportional number of negative samples, including invalid unit–unit and unit–non-unit pairs.

4.2. Prompting with self-refinement

We employ in-context learning with three widely used open-source, medium-sized LLMs: **Qwen2.5-14B-Instruct** [28], **DeepSeek-R1-Distill-Qwen-14B** [29], and **Gemma-3-12b-it** [30], accessed via Hugging Face. Prompts are designed for both *single-step* and *multi-step* settings using meta-prompting techniques [31], with argument schema constraints embedded in the system message (see Appendix B). Final templates are selected empirically and adapted to each model's chat-completion format. In the **single-step** setting, models generate full argument structures in one pass, whereas in the **multi-step** setting, the process is decomposed into unit extraction followed by relation prediction (*support*/*attack*) between the predicted units.

To enforce schema compliance, outputs are validated using Pydantic [32]. Invalid outputs trigger a single retry, where validation errors and corrective instructions are injected into the prompt automatically, following self-refinement strategies for reducing hallucinations and incomplete outputs [33].

To study supervision effects, we vary in-context examples (zero-, three-, and five-shot), sampled from

QT30 training data with balanced support and attack coverage. Examples include brief explanations of each unit's dialogical role, inspired by speech-act information in IAT annotations (Figure 1). Models are prompted to generate such explanations to support more grounded unit and relation predictions.

5. Experimental Setup

The experimental setup describes training and inference for both fine-tuned and prompting approaches (Table 2). All experiments have been conducted on a single 48GB NVIDIA Ampere A40 GPU.

Fine-tuning Language Models. The single- and multi-step models were fine-tuned using the QT30 training corpus. Hyperparameter optimization was performed using Optuna [34], with eight trials per configuration to select the learning rate from a log-uniform range of $[1e{-}5, 5e{-}4]$. Batch size and dropout were fixed to 8 and 0.1. To address label imbalance across subtasks, weighted loss functions were applied. Using the best hyperparameters, models were then re-trained using 5-fold cross-validation with early stopping based on evaluation loss (patience = 3), ensuring a stable estimate of performance and a robust model selection criterion.

Inference with Finetuned Models. Predictions were generated using the best-performing models and post-processed according to the schema validation logic (Section 3). In single-step, this involved correcting BIO predictions to resolve span inconsistencies and relational distance conflicts. In the multi-step, malformed spans (those containing an errored word or only punctuation) were filtered and chronologically valid unit pairs were constructed before relation prediction. The final argument structure consists of schema-compliant units with validated support and attack relations.

Inference with Prompted LLMs. LLM inference was performed using the Outlines Python library [35], which enables structured JSON generation and integrates with our Pydantic-based schema validation (Section 4.2). Decoding parameters were fixed to temperature 0, top_p 0.0, random seed 42, and repetition penalty 1.0 to ensure deterministic outputs and improve schema adherence. Maximum **two** tries per conversation are allowed.

6. Results

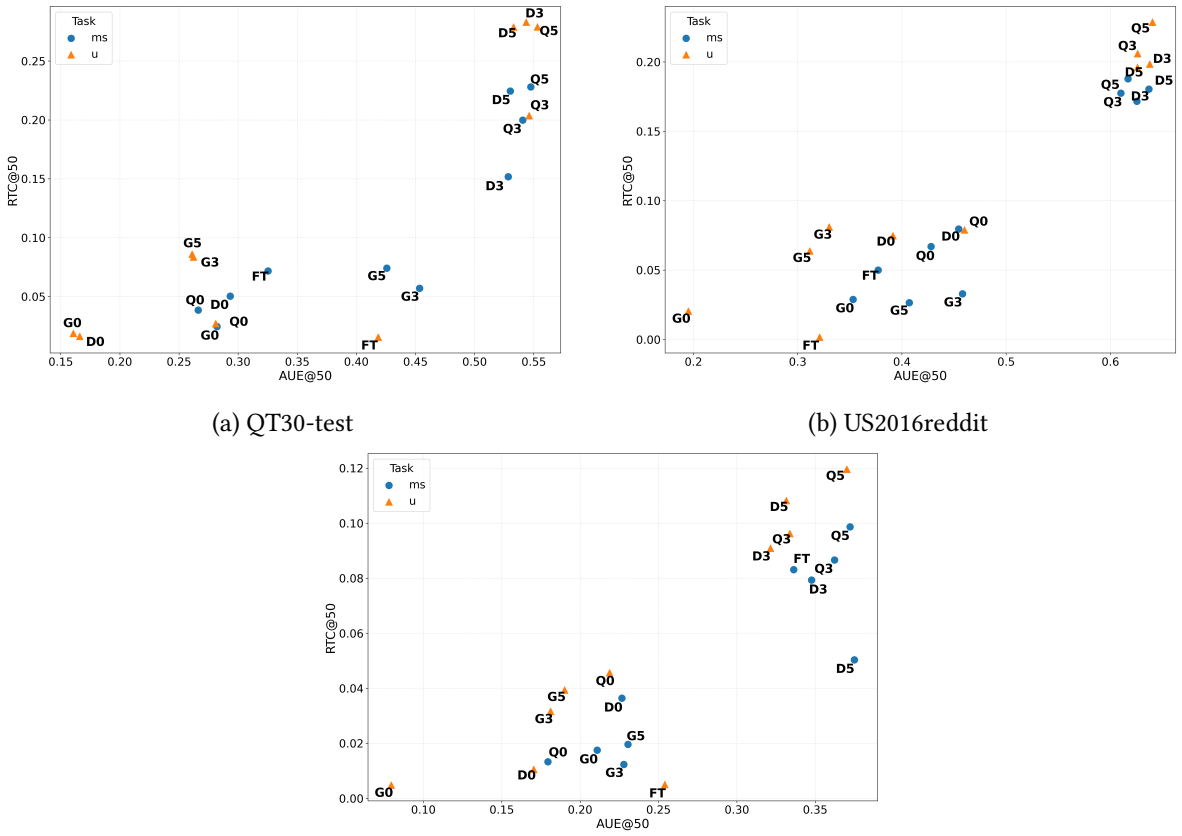
A prediction is considered a valid generation if it satisfies the schema (see Section 3), thereby avoiding malformed/empty outputs. Minor violations (e.g., single-word units or invalid IDs) are corrected via post-processing, while major inconsistencies in schema result in invalid predictions.

All models are evaluated on three datasets: QT30-test (in-domain), US2016reddit (cross-domain), and RIP1 (cross-domain). We assess predictive performance, cross-domain generalization, computational efficiency, and schema compliance.

We follow the evaluation protocol of [2], which decomposes ASP into two phases: argument unit evaluation and relation evaluation.

Argument units are evaluated at the span level by aligning predicted and gold spans using a one-to-one best-alignment procedure under two span overlap thresholds: 50% partial overlap and 100% exact match [2]. A predicted span is considered correct if it is aligned to a gold span whose overlap exceeds the threshold. We measure precision, recall, and F1 for the ARG and NON-ARG class, counting unaligned predicted spans as false positives and unaligned gold spans as false negatives.

Relation evaluation begins with first assessing whether aligned units are paired with their correct counterparts. For correctly aligned and paired units, we then evaluate whether the predicted relation type (`support` or `attack`) between them is correct. A relation is counted as a true positive if (i) the source and target of predicted pair are aligned, (ii) the predicted pair also exists in the gold structure, and (iii) the relation type is correct, otherwise noted as false . Relation-level precision, recall, and F1 are measured under both span-overlap thresholds.



(a) QT30-test

(b) US2016reddit

(c) RIP1

Figure 2: Task performance across datasets and configurations (identified by Model Codes [MC]; see Table 2) for single- (*ss*) and multi-step (*ms*) settings. *Plots (a–c)* show argument unit extraction (x: AUE@50) and relation classification (y: RTC@50) at 50% span overlap, with each point representing the weighted F1 score (AUE, RTC).

6.1. Task Performance and Generalization

To analyze model and architectural influence on ASP performance, we present corpus-wise scatter plots in Figure 2. The x-axis represents weighted F1 for argument unit extraction (AUE) subtask, while the y-axis represents weighted F1 score for relation type classification (RTC) subtask, both evaluated under the 50% span-overlap condition.

In-domain results on QT30-test (Figure 2a) show that single-step architecture achieve stronger RTC performance than multi-step architecture though they perform similarly on AUE. Relation prediction benefits from the joint modeling.

Prompt-based models with more context examples (Q3, 5, D3, 5) consistently outperform fine-tuned models, while zero-shot and Gemma-based configurations perform the poorest. This indicates that adding context examples facilitates both the subtasks significantly. However, this behaviour is not uniform across the different LLM families. Qwen2.5-14B-Instruct demonstrated the best performance due to better schema compliance while Gemma3-12B-it suffered from poor schema compliance that resulted in malformed generations. We discuss this further in the next section.

Studying the cross-domain results over the other two corpora, US2016reddit and RIP1 (Figure 2b–c) shows a significant performance drop for RTC than AUE indicating that relations do not generalize well between the different corpora. Single-step architectures continue to outperform the multi-step for RTC over AUE indicating that RTC does benefit from the joint modeling. Performance is higher on US2016reddit than on RIP1, likely due to US2016reddit having closer alignment with QT30 in annotation and discourse characteristics. RIP1 conversations are generally longer, have sparser distribution of units and relations, and are non-political and collaborative in nature. These distinct characteristic differences

	QT30-test (168 conv.)			**RIP1** (79 conv.)		
MC	SR(%)	T(s)	Tk	SR(%)	T(s)	Tk
$Q0_{ss}$	$96.4_{+3.6}$	$27.46_{+0.6}$	$210k_{+13k}$	$100.0_{+0.0}$	$37.38_{+0.0}$	$119k_{+0k}$
$Q3_{ss}$	$100.0_{+0.0}$	$34.49_{+0.0}$	$623k_{+0}$	$98.7_{+1.3}$	$34.45_{+0.0}$	$301k_{+8k}$
$Q5_{ss}$	$100.0_{+0.0}$	$37.89_{+0.0}$	$830k_{+0}$	$98.7_{+1.3}$	$37.3_{+0.3}$	$397k_{+10k}$
$D3_{ss}$	$100.0_{+0.0}$	$46.11_{+0.0}$	$646k_{+0}$	$97.4_{+0.0}$	$48.78_{+2.0}$	$312k_{+4k}$
$D5_{ss}$	$100.0_{+0.0}$	$47.76_{+0.0}$	$847k_{+0}$	$97.4_{+0.0}$	$52.18_{+2.0}$	$405k_{+4k}$
$G5_{ss}$	$35.7_{+22.0}$	132.46_{+90}	$529k_{+529k}$	$19.0_{+20.3}$	149.7_{+120}	$210k_{+270k}$
$Q0_{ms}$	$98.8_{+1.2}$	$18.6_{+0.0}$	$284k_{+4k}$	$98.7_{+1.3}$	$16.6_{+0.0}$	$147k_{+2k}$
$Q3_{ms}$	$99.4_{+0.6}$	$32.55_{+0.0}$	$1049k_{+8k}$	$100.0_{+0.0}$	$27.14_{+0.0}$	$503k_{+0}$
$Q5_{ms}$	$100.0_{+0.0}$	$35.27_{+0.0}$	$1416k_{+0}$	$100.0_{+0.0}$	$31.02_{+0.0}$	$676k_{+0}$
$D3_{ms}$	$95.8_{+4.2}$	$42.57_{+1.0}$	$1069k_{+49k}$	$79.74_{+18.98}$	$38.04_{+2.5}$	$493k_{+78k}$
$D5_{ms}$	$50.0_{+50.0}$	$40.27_{+3.0}$	$1203k_{+263k}$	$15.2_{+78.5}$	$32.43_{+5.0}$	$476k_{+262k}$
$G5_{ms}$	$11.9_{+80.9}$	$111.58_{+63.0}$	$948k_{+621k}$	$12.6_{+78.5}$	$118.01_{+60.0}$	$448k_{+349k}$

Table 3
Computational efficiency of configurations across QT30-test and RIP1. SR: success rate of schema-compliant predictions in first try; T: Average time taken per conversation in first try; Tk: total tokens (in thousands) outputted for all conversations in first try. Subscripts (+x) denote changes observed when a retry is performed.

likely resulted in the significant performance drop compared to the other two corpora.

Our findings show that prompt-based single-step configurations performed the best among all the tested configurations. However, none of the configurations generalized well to unseen corpora or different dialogical setting.

6.2. Computational Efficiency and Schema Compliance

While prompting generally outperforms fine-tuning, it incurs higher computational cost and more frequent schema violations (malformed/empty outputs), likely due to hallucinated outputs and the difficulty of generating complete schema-compliant outputs with reasoning enabled.

Minor violations, such as poorly segmented units (1–2 words or invalid IDs), are corrected through post-processing, while major violations (e.g., incomplete JSON structures) trigger a single retry via self-refinement. In this step, the model is re-prompted with error reports and corrective feedback. While this improves schema compliance, it increases runtime and token usage. Table 3 reports first-pass success rates for generating schema-compliant structures, along with average inference time and total token consumption, and quantifies the additional cost introduced by retries.

Our findings show that effectiveness of the retry mechanism is strongly dependent on task architecture: multi-step settings benefit more, whereas single-step already achieve high first-try performance and exhibit only marginal gains if retry was triggered. Analysis of multi-step predictions indicates that retries are triggered more frequently during the argument unit extraction (AUE) stage, largely due to over-prediction of units. In contrast, single-step architectures may benefit from more grounded predictions through joint unit–relation modeling.

Retries are most beneficial in low-context settings (e.g., zero-shot), where they yield higher gains per additional token. However, they substantially increase computational cost—often nearly doubling token usage—while providing limited overall improvements.

Overall, these results highlight a trade-off between generating complex structured outputs and maintaining computational efficiency. Future work should explore selective retry strategies, such as triggering retries based on confidence estimates or restricting them to low-context settings, to better balance this trade-off.

	QT30-test		US2016reddit		RIP1	
MC	$AUE_{P/R}$	$RTC_{P/R}$	$AUE_{P/R}$	$RTC_{P/R}$	$AUE_{P/R}$	$RTC_{P/R}$
FT_{ms}	0.32 / 0.35	0.12 / 0.06	0.42 / 0.36	0.08 / 0.04	0.28 / **0.44**	0.12 / 0.08
$Q0_{ss}$	0.35 / 0.25	0.06 / 0.02	0.55 / 0.42	0.18 / 0.05	0.21 / 0.25	0.12 / 0.03
$Q3_{ss}$	0.52 / 0.61	0.33 / 0.16	0.64 / 0.64	0.35 / 0.16	0.32 / 0.38	0.25 / 0.06
$Q5_{ms}$	**0.53** / 0.61	0.38 / 0.18	**0.67** / 0.60	0.30 / 0.14	**0.38** / 0.38	0.24 / 0.06
$Q5_{ss}$	0.51 / **0.64**	**0.44** / 0.22	0.65 / 0.65	**0.41** / **0.17**	0.34 / 0.43	**0.27** / **0.08**
$D5_{ms}$	0.46 / 0.68	0.32 / 0.19	0.62 / **0.69**	0.25 / 0.15	0.30 / 0.40	0.19 / 0.06
$D5_{ss}$	0.45 / 0.69	0.40 / **0.24**	0.59 / 0.69	0.29 / 0.14	0.28 / **0.44**	0.21 / 0.08

Table 4
Performance analysis across model configurations (MC, see Table 2). Precision (P) and recall (R) are reported as P / R for argument unit extraction (AUE) and relation classification (RTC). Bold values indicate the best performance within each dataset.

6.3. Error Analysis

We conducted an error analysis of the model predictions for our best performing configurations and the three corpora. We study three main stages in ASP here: the main failure modes in AUE subtask, and the RTC subtask split into relation prediction (source-target unit pairing) and type classification (support–attack).

Analyzing the predicted argument units obtained from AUE, the primary errors occurred due to segmentation inconsistencies, i.e., the models failed to segment and extract units from the raw text efficiently. More often, the LLM-based models tend to over-segment the text resulting in poor precision. Adopting a 50% partial overlap allowed for better unit alignment with the gold units and resulted in better recall (Table 4). However, at 100% perfect overlap, the performance drastically falls indicating that the models struggle to efficiently identify segment boundaries. Explicit segmentation with clear rules is needed for improving performance in argument unit prediction and extraction, especially in dialogical settings where statements are often incomplete or poorly-constructed making argument boundaries difficult to be assessed.

Analysing the downstream relation prediction, one of the main failure modes is the partially-overlapped units getting incorrectly paired as source-targets. This led to *endpoint misalignment* with the gold pairs (when either the source or target fails to align with a gold unit) and made the relation in-valid. Additionally, most LLM-based approaches behaved conservatively and predicted few relations, that led to higher number of missing relations. These two errors resulted in significantly poor recall as most relations were missing or marked missing due to in-validity for evaluation (Table 4). Since only correctly paired relations proceeded for evaluation, precision was relatively stable.

Through manual inspection, two factors are identified: (i) limited contextual understanding and (ii) long-distance dependencies. Operating in a multi-party dialogical settings that contain implicit and noisy instances, models often failed to accurately resolve the implicitness in the discussion based on the limited context provided. For instance, linking "You are wrong" to "he's killing it" requires broader dialogical context from subsequent turns (see Figure 1). Single-step configurations, especially prompt-based configurations, may be less susceptible to this since they model the whole conversation directly. More work is needed to investigate this further. Additionally, in RIP1 corpus and US2016reddit corpus, that have relatively longer discussions and long-distance argumentative interactions (Table 1), configurations failed to identify these connections and mostly predicted relations between immediate neighbours.

Finally, analyzing the correctly paired relations for type classification, most configurations performed well, achieving near-perfect *support* and good *attack* scores despite the inherent class imbalance. This indicates that relation pairing, rather than type classification, is the primary bottleneck. More details on the quantitative error analysis is provided in Appendix C. Future work must decouple the relation prediction and classification type, i.e., link argument units as potential source–target pairs explicitly

before predicting the nature of the relation itself.

7. Conclusion

In this work, we address ASP for online conversations. We adapt diverse IAT-annotated corpora into bipolar argument structures enabling their use for computational ASP. Building on this representation, we conduct a systematic evaluation of ASP, comparing supervised fine-tuning and prompt-based LLMs across single- and multi-step task architectures. Our analysis examines the effect of model design choices on performance, generalization, schema compliance, and computational efficiency under shared structural constraints. We release our data processing and modeling pipelines, which supports IAT-based corpora in AIF format and can be extended to a wide range of language models for dialogical argumentation (Appendix A).

Our findings find that predicting argument structures from online conversations remains a challenging task. Testing various modeling configurations over diverse dialogical datasets demonstrates that the key bottleneck in ASP remains in identifying if two given argument units are argumentatively linked or not. This challenge is further amplified in dialogical settings that contain implicit or context-dependent argumentative content. It requires broader discourse context and deeper reasoning to infer argumentative relations accurately. Our results indicate the need for explicit contextual grounding when designing ASP systems for complex dialogical interactions.

A limitation of our work has been the limited modeling strategies that have been explored. Future work should extend this analysis to include fine-tuning LLMs [36], testing graph-based neural networks [37], and adopting neuro-symbolic frameworks for contextual grounding of both arguments and their relations [38]. Another limitation is the limited cross-domain generalization of the evaluated ASP approaches, as performance degrades on datasets with distinct dialogical and argumentative characteristics. Moreover, some of the datasets used in this study may have appeared in the pre-training corpora of the evaluated LLMs. Future work should evaluate ASP on a broader and more diverse set of dialogical corpora to improve robustness to out-of-distribution data and better assess model generalization. Given the limited availability of annotated resources for dialogical settings, we hope that our data processing pipeline will facilitate future research and advance the computational modeling of dialogical ASP.

Finally, one of the main challenges to dialogical ASP is relation prediction, specifically identifying the source–target pair. Future research should focus on enhancing contextual reasoning such as supplementing contextual knowledge to an argument using retrieval augmented generation (RAG) [39], or incorporating structural knowledge to improve relation prediction directly [1].

Developing robust and reliable dialogical ASP systems can support real-world applications that require argumentation comprehension such as in conflict resolution, collaborative reasoning and decision-making, and argument-based summarization and explanations.

8. Acknowledgments

This research work has received funding from the European Union's Horizon Europe research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 101073351. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Research Executive Agency (REA). Neither the European Union nor the granting authority can be held responsible for them.

Declaration on Generative AI

During the preparation of this work,we used ChatGPT in order to assist with grammar, formal tone, and spelling check. We have carefully reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] D. Dore, S. Faralli, S. Villata, Leveraging Graph Structural Knowledge to Improve Argument Relation Prediction in Political Debates, in: E. Chistova, P. Cimiano, S. Haddadan, G. Lapesa, R. Ruiz-Dolz (Eds.), Proceedings of the 12th Argument Mining Workshop, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 74–86. doi:10.18653/v1/2025.argmining-1.7.

[2] M. T. Sazid, R. E. Mercer, A Unified Representation and a Decoupled Deep Learning Architecture for Argumentation Mining of Students' Persuasive Essays, in: Proceedings of the 9th Workshop on Argument Mining, International Conference on Computational Linguistics, Online and in Gyeongju, Republic of Korea, 2022, pp. 74–83.

[3] G. Morio, H. Ozaki, T. Morishita, K. Yanai, End-to-end Argument Mining with Cross-corpora Multi-task Learning, Transactions of the Association for Computational Linguistics 10 (2022) 639–658. doi:10.1162/tacl_a_00481.

[4] P. Santin, G. Grundler, A. Galassi, F. Galli, F. Lagioia, E. Palmieri, F. Ruggeri, G. Sartor, P. Torroni, Argumentation Structure Prediction in CJEU Decisions on Fiscal State Aid, in: Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law, ACM, Braga Portugal, 2023, pp. 247–256. doi:10.1145/3594536.3595174.

[5] J. Cabessa, H. Hernault, U. Mushtaq, Argument Mining in BioMedicine: Zero-Shot, In-Context Learning and Fine-tuning with LLMs, in: Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024), CEUR Workshop Proceedings, Pisa, Italy, 2024, pp. 122–131.

[6] A. Farzam, S. Shekhar, I. Mehlhaff, M. Morucci, Multi-Task Learning Improves Performance in Deep Argument Mining Models, in: Proceedings of the 11th Workshop on Argument Mining (ArgMining 2024), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 46–58. doi:10.18653/v1/2024.argmining-1.5.

[7] M. Fromm, E. Faerman, M. Berrendorf, S. Bhargava, R. Qi, Y. Zhang, L. Dennert, S. Selle, Y. Mao, T. Seidl, Argument Mining Driven Analysis of Peer-Reviews, Proceedings of the AAAI Conference on Artificial Intelligence 35 (2021) 4758–4766. doi:10.1609/aaai.v35i6.16607.

[8] A. Hautli-Janisz, Z. Kikteva, W. Siskou, K. Gorska, R. Becker, C. Reed, QT30: A Corpus of Argument and Conflict in Broadcast Debate, in: Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 3291–3300.

[9] E. Schad, J. Visser, C. Reed, The RIP Corpus of Collaborative Hypothesis-Making, in: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, 2024, pp. 16047–16057.

[10] S. Saha, S. Das, R. Srihari, EDU-AP: Elementary Discourse Unit based Argument Parser, in: Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue, Association for Computational Linguistics, Edinburgh, UK, 2022, pp. 183–192. doi:10.18653/v1/2022.sigdial-1.19.

[11] J. Lawrence, C. Reed, Argument Mining: A Survey, Computational Linguistics 45 (2020) 765–818. doi:10.1162/coli_a_00364.

[12] J. Cabessa, H. Hernault, U. Mushtaq, Argument mining in biomedicine: Zero-shot, in-context learning and fine-tuning with llms, in: Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024), CEUR Workshop Proceedings, 2024, pp. 122–131. URL: https://aclanthology.org/2024.clicit-1.16/.

[13] D. Gorur, A. Rago, F. Toni, Can large language models perform relation-based argument mining?, in: Proceedings of the 31st International Conference on Computational Linguistics (COLING 2025), Association for Computational Linguistics, 2025, pp. 8518–8534. URL: https://aclanthology.org/2025.coling-main.569/. doi:10.18653/v1/2025.coling-main.569.

[14] I. J. Cruickshank, L. H. X. Ng, Prompting and Fine-Tuning Open-Sourced Large Language Models for Stance Classification, ACM Trans. Intell. Syst. Technol. (2025). doi:10.1145/3725816.

[15] J. Lawrence, F. Bex, C. Reed, M. Snaith, AIFdb: Infrastructure for the Argument Web, in: Computational Models of Argument, IOS Press, 2012, pp. 515–516. doi:10.3233/

978-1-61499-111-3-515.
[16] K. Budzynska, C. Reed, Whence inference, University of Dundee Technical Report (2011).
[17] C. Stab, I. Gurevych, Identifying Argumentative Discourse Structures in Persuasive Essays, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 46–56. doi:10.3115/v1/D14-1006.
[18] Y. Ye, S. Teufel, 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: Main Volume, Association for Computational Linguistics, Online, 2021, pp. 669–678. doi:10.18653/v1/2021.eacl-main.55.
[19] J. Bao, Y. He, Y. Sun, B. Liang, J. Du, B. Qin, M. Yang, R. Xu, A Generative Model for End-to-End Argument Mining with Reconstructed Positional Encoding and Constrained Pointer Mechanism, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 10437–10449. doi:10.18653/v1/2022.emnlp-main.713.
[20] L. Shi, F. Giunchiglia, H. Wang, Y. Cheng, R. Song, D. Shi, X. Diao, H. Xu, From text mining to intelligent debate: Task frameworks and technological evolution in computational argumentation, Information Processing & Management 63 (2026) 104465. doi:10.1016/j.ipm.2025.104465.
[21] H. Nguyen, D. Litman, Context-aware Argumentative Relation Mining, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 1127–1137. doi:10.18653/v1/P16-1107.
[22] M. Feger, K. Boland, S. Dietze, Limited Generalizability in Argument Mining: State-Of-The-Art Models Learn Datasets, Not Arguments, in: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vienna, Austria, 2025, pp. 23900–23915. doi:10.18653/v1/2025.acl-long.1164.
[23] I. Rahwan, C. Reed, The Argument Interchange Format, in: Argumentation in Artificial Intelligence, Springer US, Boston, MA, 2009, pp. 383–402. doi:10.1007/978-0-387-98197-0_19.
[24] N. Potyka, Bipolar Abstract Argumentation with Dual Attacks and Supports, Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning 17 (2020) 677–686. doi:10.24963/kr.2020/69.
[25] J. Visser, B. Konat, R. Duthie, M. Koszowy, K. Budzynska, C. Reed, Argumentation in the 2016 US presidential elections: Annotated corpora of television debates and social media reaction, Language Resources and Evaluation 54 (2020) 123–154. doi:10.1007/s10579-019-09446-8.
[26] M. Sagen, Large-Context Question Answering with Cross-Lingual Transfer, Master's thesis, Uppsala University, Department of Information Technology, 2021. 45 pages.
[27] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, CoRR abs/1911.02116 (2019). arXiv:1911.02116.
[28] Qwen, A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, Z. Qiu, Qwen2.5 Technical Report, 2025. doi:10.48550/arXiv.2412.15115. arXiv:2412.15115.
[29] D. Guo, D. Yang, H. Zhang, J. Song, P. Wang, Q. Zhu, R. Xu, R. Zhang, S. Ma, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, B. Wang, B. Wu, B. Feng, C. Lu, C. Zhao, C. Deng, C. Ruan, D. Dai, D. Chen, D. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Xu, H. Ding, H. Gao, H. Qu, H. Li, J. Guo, J. Li, J. Chen, J. Yuan, J. Tu, J. Qiu, J. Li, J. L. Cai, J. Ni, J. Liang, J. Chen, K. Dong, K. Hu, K. You, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Zhao, L. Wang, L. Zhang, L. Xu, L. Xia, M. Zhang, M. Zhang, M. Tang, M. Zhou, M. Li, M. Wang, M. Li, N. Tian, P. Huang, P. Zhang, Q. Wang, Q. Chen, Q. Du, R. Ge, R. Zhang, R. Pan, R. Wang, R. J. Chen, R. L. Jin, R. Chen, S. Lu, S. Zhou, S. Chen, S. Ye, S. Wang, S. Yu, S. Zhou, S. Pan,

S. S. Li, S. Zhou, S. Wu, T. Yun, T. Pei, T. Sun, T. Wang, W. Zeng, W. Liu, W. Liang, W. Gao, W. Yu, W. Zhang, W. L. Xiao, W. An, X. Liu, X. Wang, X. Chen, X. Nie, X. Cheng, X. Liu, X. Xie, X. Liu, X. Yang, X. Li, X. Su, X. Lin, X. Q. Li, X. Jin, X. Shen, X. Chen, X. Sun, X. Wang, X. Song, X. Zhou, X. Wang, X. Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. Zhang, Y. Xu, Y. Li, Y. Zhao, Y. Sun, Y. Wang, Y. Yu, Y. Zhang, Y. Shi, Y. Xiong, Y. He, Y. Piao, Y. Wang, Y. Tan, Y. Ma, Y. Liu, Y. Guo, Y. Ou, Y. Wang, Y. Gong, Y. Zou, Y. He, Y. Xiong, Y. Luo, Y. You, Y. Liu, Y. Zhou, Y. X. Zhu, Y. Huang, Y. Li, Y. Zheng, Y. Zhu, Y. Ma, Y. Tang, Y. Zha, Y. Yan, Z. Z. Ren, Z. Ren, Z. Sha, Z. Fu, Z. Xu, Z. Xie, Z. Zhang, Z. Hao, Z. Ma, Z. Yan, Z. Wu, Z. Gu, Z. Zhu, Z. Liu, Z. Li, Z. Xie, Z. Song, Z. Pan, Z. Huang, Z. Xu, Z. Zhang, Z. Zhang, DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning, Nature 645 (2025) 633–638. doi:10.1038/s41586-025-09422-z.

[30] G. Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin, T. Matejovicova, A. Ramé, M. Rivière, L. Rouillard, T. Mesnard, G. Cideron, J.-b. Grill, S. Ramos, E. Yvinec, M. Casbon, E. Pot, I. Penchev, G. Liu, F. Visin, K. Kenealy, L. Beyer, X. Zhai, A. Tsitsulin, R. Busa-Fekete, A. Feng, N. Sachdeva, B. Coleman, Y. Gao, B. Mustafa, I. Barr, E. Parisotto, D. Tian, M. Eyal, C. Cherry, J.-T. Peter, D. Sinopalnikov, S. Bhupatiraju, R. Agarwal, M. Kazemi, D. Malkin, R. Kumar, D. Vilar, I. Brusilovsky, J. Luo, A. Steiner, A. Friesen, A. Sharma, A. Sharma, A. M. Gilady, A. Goedeckemeyer, A. Saade, A. Feng, A. Kolesnikov, A. Bendebury, A. Abdagic, A. Vadi, A. György, A. S. Pinto, A. Das, A. Bapna, A. Miech, A. Yang, A. Paterson, A. Shenoy, A. Chakrabarti, B. Piot, B. Wu, B. Shahriari, B. Petrini, C. Chen, C. L. Lan, C. A. Choquette-Choo, C. J. Carey, C. Brick, D. Deutsch, D. Eisenbud, D. Cattle, D. Cheng, D. Paparas, D. S. Sreepathihalli, D. Reid, D. Tran, D. Zelle, E. Noland, E. Huizenga, E. Kharitonov, F. Liu, G. Amirkhanyan, G. Cameron, H. Hashemi, H. Klimczak-Plucińska, H. Singh, H. Mehta, H. T. Lehri, H. Hazimeh, I. Ballantyne, I. Szpektor, I. Nardini, J. Pouget-Abadie, J. Chan, J. Stanton, J. Wieting, J. Lai, J. Orbay, J. Fernandez, J. Newlan, J.-y. Ji, J. Singh, K. Black, K. Yu, K. Hui, K. Vodrahalli, K. Greff, L. Qiu, M. Valentine, M. Coelho, M. Ritter, M. Hoffman, M. Watson, M. Chaturvedi, M. Moynihan, M. Ma, N. Babar, N. Noy, N. Byrd, N. Roy, N. Momchev, N. Chauhan, N. Sachdeva, O. Bunyan, P. Botarda, P. Caron, P. K. Rubenstein, P. Culliton, P. Schmid, P. G. Sessa, P. Xu, P. Stanczyk, P. Tafti, R. Shivanna, R. Wu, R. Pan, R. Rokni, R. Willoughby, R. Vallu, R. Mullins, S. Jerome, S. Smoot, S. Girgin, S. Iqbal, S. Reddy, S. Sheth, S. Põder, S. Bhatnagar, S. R. Panyam, S. Eiger, S. Zhang, T. Liu, T. Yacovone, T. Liechty, U. Kalra, U. Evci, V. Misra, V. Roseberry, V. Feinberg, V. Kolesnikov, W. Han, W. Kwon, X. Chen, Y. Chow, Y. Zhu, Z. Wei, Z. Egyed, V. Cotruta, M. Giang, P. Kirk, A. Rao, K. Black, N. Babar, J. Lo, E. Moreira, L. G. Martins, O. Sanseviero, L. Gonzalez, Z. Gleicher, T. Warkentin, V. Mirrokni, E. Senter, E. Collins, J. Barral, Z. Ghahramani, R. Hadsell, Y. Matias, D. Sculley, S. Petrov, N. Fiedel, N. Shazeer, O. Vinyals, J. Dean, D. Hassabis, K. Kavukcuoglu, C. Farabet, E. Buchatskaya, J.-B. Alayrac, R. Anil, Dmitry, Lepikhin, S. Borgeaud, O. Bachem, A. Joulin, A. Andreev, C. Hardin, R. Dadashi, L. Hussenot, Gemma 3 Technical Report, 2025. doi:10.48550/arXiv.2503.19786. arXiv:2503.19786.

[31] Y. Hou, H. Dong, X. Wang, B. Li, W. Che, MetaPrompting: Learning to learn better prompts, in: Proceedings of the 29th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2022, pp. 3251–3262.

[32] S. Colvin, E. Jolibois, H. Ramezani, A. Garcia Badaracco, T. Dorsey, D. Montague, S. Matveenko, M. Trylesinski, S. Runkle, D. Hewitt, A. Hall, V. Plot, Pydantic Validation, 2026. URL: https://github.com/pydantic/pydantic.

[33] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, S. Gupta, B. P. Majumder, K. Hermann, S. Welleck, A. Yazdanbakhsh, P. Clark, SELF-REFINE: Iterative refinement with self-feedback, in: Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23, Curran Associates Inc., Red Hook, NY, USA, 2023, pp. 46534–46594.

[34] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019. URL: https://optuna.org/.

[35] B. T. Willard, R. Louf, Efficient guided generation for large language models, arXiv preprint

arXiv:2307.09702 (2023).
[36] S. Efeoglu, A. Paschke, Fine-Tuning Large Language Models for Relation Extraction within a Retrieval-Augmented Generation Framework, in: H. Fei, K. Tu, Y. Zhang, X. Hu, W. Han, Z. Jia, Z. Zheng, Y. Cao, M. Zhang, W. Lu, N. Siddharth, L. Øvrelid, N. Xue, Y. Zhang (Eds.), Proceedings of the 1st Joint Workshop on Large Language Models and Structure Modeling (XLLM 2025), Association for Computational Linguistics, Vienna, Austria, 2025, pp. 1–7. doi:10.18653/v1/2025.xllm-1.1.
[37] Y. Sun, J. Bao, G. Tu, B. Liang, M. Yang, R. Xu, Multi-view Hierarchical Graph Neural Network for Argumentation Mining, Cognitive Computation 17 (2024) 31. doi:10.1007/s12559-024-10391-0.
[38] M. Plenz, P. Heinisch, A. Frank, P. Cimiano, PAKT: Perspectivized Argumentation Knowledge Graph and Tool for Deliberation Analysis, in: Robust Argumentation Machines, Springer Nature Switzerland, Cham, 2024, pp. 89–107. doi:10.1007/978-3-031-63536-6_6.
[39] W. Sun, M. Li, J. Davis, E. Cabrio, S. Villata, M.-F. Moens, Weakly-supervised Argument Mining with Boundary Refinement and Relation Denoising, in: M. Akhtar, R. Aly, R. Cao, C. Christodoulopoulos, O. Cocarascu, Z. Guo, A. Mittal, M. Schlichtkrull, J. Thorne, A. Vlachos (Eds.), Proceedings of the Ninth Fact Extraction and VERification Workshop (FEVER), Association for Computational Linguistics, Rabat, Morocco, 2026, pp. 1–12.

A. Online Resources

The data and system code are available via

- Github: System repository
- Github: Data Processing Pipeline Repository
- Data corpora: IAT2BAS Dataset

B. Prompts used for ASP

We provide the templates for the three prompts used in this study here in Figures 3, 4, and 5. More details on their use can be viewed in the Github repository.

C. Quantitative Error Analysis Details

We observe six types of structural errors in the datasets and across configurations. False positive units are predicted units that do not exist in the ground truth structures, likely due to segmentation errors. False negative units are non-predicted units that exist in ground truth structures but were not predicted by the models. These are observed to be more implicit arguments that require human-level reasoning and contextual understanding to be identified.

In terms of relation errors, we identify four types. Endpoint mis-matched relations are relations that exist either in ground truth or in predicted structure but can not be evaluated because either source or the target unit is false, i.e., all relations that are linked to false positive or false negative units are automatically categorized as endpoint mis-matched relations by us. False positive relations are predicted relations between two true positive units that do not exist in ground truth structure. False negative relations are non-predicted relations between two true positive units that do exist in ground truth structure but missing in predicted structures. Finally, mis-classified relations are relations that have been correctly identified between two true positive units but has been mis-labelled (incorrect support or attack label). Figure 6 presents a visualization of the six structural errors.

In Table 5 we list the quantitative values of the six structural errors observed in the structure across the datasets and the configuration. We observe that for QT30 and RIP1 corpora, LLM configurations are generally more liberal, overly predicting units, leading to higher false-positive units and relatively

Prompt for Single-Step ASP

```
You are an argument mining expert. From the discussion, extract argument units
and their relations.

Return ONE JSON object with this exact shape:

{ "argument_units": [ {"reason": "...", "id": 0, "text": "..."}, ... ],
"relations": [ {"source_id": 1, "target_id": 0, "type": "support"}, ... ] }

Argument units:

  • text: copy verbatim from the discussion; do not paraphrase.
  • Each unit is a single argumentative idea, e.g., asserts, questions,
    rejects, accepts, defends, or challenges a topic or another statement.
  • Assign IDs in order of appearance: 0, 1, 2, . . .
  • reason: short explanation of the unit's intent or expressed idea.

Relations:

  • Only create a relation if there is a clear indication in the conversation
    text.
  • Only treat a pair as "no relation" if they are clearly unrelated or only
    share a topic without one supporting or attacking the other.
  • support: source accepts or gives reasons, evidence, or clarification for
    target.
  • attack: source challenges, rejects, undercuts, or undermines target.
  • Only create a relation if there is a clear, explicit link, e.g., "because",
    "but", "however", or "in response to".
  • If unsure or only loosely related by topic, do not create a relation.
  • Use ONLY IDs from argument_units; never invent new IDs.
  • By default, the source should appear after the target in the discussion.
  • Do NOT create symmetric duplicates.

Schema constraints:

  • Root object MUST have ONLY: "argument_units" and "relations".
  • "argument_units" and "relations" MUST each be lists.
  • Elements of "argument_units" MUST have ONLY: "reason", "id", and "text".
  • Elements of "relations" MUST have ONLY: "source_id", "target_id", and
    "type".
  • "type" MUST be exactly "support" or "attack".
  • Do NOT output any text before or after the JSON.

Few-shot examples:

[FEW-SHOT EXAMPLE 1]
[FEW-SHOT EXAMPLE 2]

User input:

Discussion:
<conversation>
```

Figure 3: Prompt template used for single-step argument structure prediction.

lower false-negative units. But for US2016reddit the reverse behavior is observed, potentially due to this dataset emerging from social media conversations that are more argumentative and more implicit in nature.

Studying the relation errors, we do have clear indications that all LLMs across the three corpora behave very conservatively with significantly higher false negative relations than false positive relations. Another main failure mode is the endpoint mis-matched relations. Higher false-positive units or false-negative units naturally leads to higher endpoint mis-matched relations. We see the clear dependency

Prompt for Argument Unit Extraction in Multi-Step ASP

```
You are an argument mining expert. Extract argument units from the discussion.

Return ONE JSON object with this exact shape:

{ "argument_units": [ {"reason": "...", "id": 0, "text": "..."}, ... ] }

Rules:

  • Copy text verbatim from the discussion; do not paraphrase.
  • Each unit MUST be a single argumentative idea, e.g., asserts, questions,
    rejects, accepts, defends, or challenges a topic or another statement.
  • Assign IDs in order of appearance: 0, 1, 2, . . .
  • reason is a short explanation of the unit's intent or role, e.g., claim,
    premise, opinion, stance, or counterclaim.
  • Exclude non-argumentative chatter, jokes, greetings, or pure rhetoric.

Schema constraints:

  • The root object MUST have ONLY the key "argument_units".
  • Each conversation should return at least 3 units.
  • Each element MUST have ONLY "id", "text", and "reason".
  • Do NOT output any text before or after the JSON.

Few-shot examples:

[FEW-SHOT EXAMPLE 1]
[FEW-SHOT EXAMPLE 2]

User input:
<conversation>
```

Figure 4: Prompt template used for argument unit extraction.

between units and the relations where a superior unit alignment strategy allows for reduced endpoint mis-matched relations.

Prompt for Relation Prediction and Type Classification in Multi-Step ASP

```
You are an argument mining expert. Identify directional relations between the
given units.

Return ONE JSON object with this exact shape:

{ "relations": [ {"source_id": ..., "target_id": ..., "type": "..."}, ... ] }

Meaning:

  • support: source gives reasons, evidence, or clarification for target.
  • attack: source challenges, rejects, undercuts, or undermines target.

Rules to avoid spurious relations:

  • Only create a relation if there is a clear indication in the text.
  • Only treat a pair as “no relation” if they are clearly unrelated or only
    share a topic without one supporting or attacking the other.
  • Only create a relation if the text shows a clear, explicit link to its
    target unit.
  • If unsure or only loosely related by topic, do NOT create a relation.
  • Ensure that the conversation text is considered as additional context, not
    only the extracted units.

Constraints:

  • Use ONLY IDs from the provided units; never invent new IDs.
  • By default, the source should appear after the target in the discussion.
  • type MUST be exactly "support" or "attack"; no other labels are allowed.
  • The root object MUST have ONLY the key "relations".
  • "relations" MUST be a list.
  • Do NOT output any text before or after the JSON.

Few-shot examples:

[FEW-SHOT EXAMPLE 1]
[FEW-SHOT EXAMPLE 2]

User input:

Discussion:
<conversation>

Argument units:
<argument_units_json>
```

Figure 5: Prompt template used for relation prediction and relation type classification.

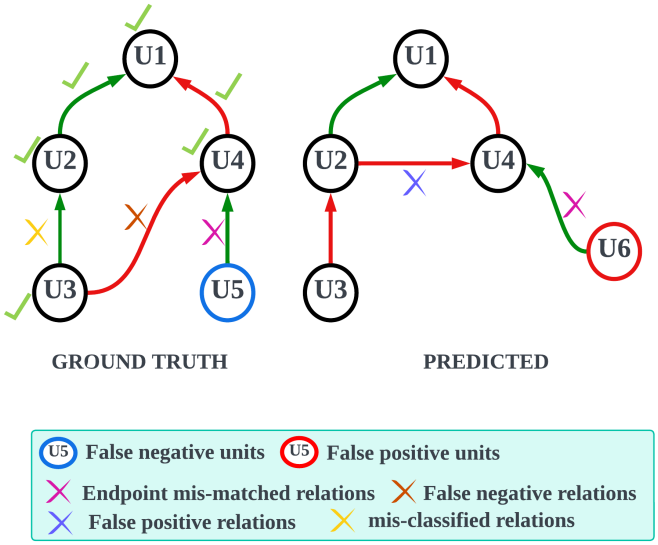


Figure 6: Visualization of the six types of structural errors observed in our error analysis between the ground truth structures and the predicted structures. The green edges represent support relations; red edges represent attack relations; U1-6 are argument units.

MC	U^+	U^-	E	R^+	R^-	C
QT30-test						
FT_{ms}	1163	1022	1836	265	934	8
$Q0_{ss}$	**647**	1178	472	71	978	3
$Q3_{ss}$	883	630	851	207	825	13
$Q5_{ms}$	853	640	799	228	812	14
$Q5_{ss}$	982	569	921	204	763	18
$D5_{ms}$	778	**297**	825	168	467	8
$D5_{ss}$	1378	490	1576	246	728	26
US2016reddit						
FT_{ms}	907	1095	2018	401	1178	32
$Q0_{ss}$	528	1026	381	131	1176	19
$Q3_{ss}$	592	667	553	267	1046	29
$Q5_{ms}$	476	732	474	312	1063	23
$Q5_{ss}$	588	630	561	250	1037	26
$D5_{ms}$	**378**	285	394	217	537	22
$D5_{ss}$	779	548	901	364	1000	51
RIP1						
FT_{ms}	864	420	1817	263	406	19
$Q0_{ss}$	723	569	378	12	448	0
$Q3_{ss}$	630	486	447	35	432	4
$Q5_{ms}$	476	473	388	51	427	6
$Q5_{ss}$	683	437	443	44	420	7
$D5_{ms}$	745	451	672	71	421	10
$D5_{ss}$	890	415	927	54	409	6

Table 5
Quantitative Error analysis across model configurations at 50% overlap threshold using seed 42 for each dataset/best configurations. U^+ and U^- denote false positive and false negative argument units. E denotes endpoint-mismatched relations. R^+ and R^- denote false positive and false negative relations. C denotes mis-classified relation types.