AN INFORMAL LOGIC LLM-BASED ARGUMENTATION FRAMEWORK

Paulo Pirozelli,* **Victor Hugo Nascimento Rocha & Fabio G. Cozman**
Universidade de São Paulo
Center for Artificial Intelligence (C4AI)

Douglas Aldred
Instituto Mauá de Tecnologia
Núcleo de Sistemas Eletrônicos Embarcados (NSEE)

ABSTRACT

Arguments are a fundamental aspect of human reasoning, in which claims are supported, challenged, and weighed against one another. We present a system that uses large language model (LLM)-based modules to reconstruct arguments from natural language text in accordance with Informal Logic (IL). The system follows a multi-step pipeline that progressively identifies argumentative components, selects relevant elements, and uncovers their logical relations. These elements are represented as a directed acyclic graph, using two component types (premises and conclusions) and three relation types (support, attack, and undercut). We conduct two complementary experiments to evaluate the system. First, we perform a manual evaluation on a dataset of arguments drawn from an IL textbook, enabling a detailed analysis of the system's ability to capture IL-style argument structures. Second, we carry out a quantitative evaluation on benchmark datasets, allowing comparison with prior work by mapping our outputs to established annotation schemes. Results show that the system is capable of recovering a range of IL-style arguments and, when adapted to different annotation schemes, achieves reasonable performance across benchmark datasets, highlighting the potential of LLM-based pipelines for scalable argument reconstruction.

1 INTRODUCTION

Arguments are developed to justify decisions, reason through problems, and persuade others of particular stances (Eemeren, 2018). When making an argument, a person offers reasons to support or attack a claim, with the intention of providing a rational justification for accepting or rejecting it. By an argument, we mean the product of this process: a set of propositions in which one of them (the conclusion) is claimed to be supported or challenged by the others. These propositions provide reasons intended to justify or refute a conclusion that is not immediately evident (Hoffmann & Catrambone, 2023).

Given their central role in reasoning and decision-making, it is not surprising that artificial intelligence (AI) has long sought to model and assess arguments computationally. Traditionally, this goal has relied on *argumentation frameworks* (AFs) (Dung, 1995), which provide formal representations for determining whether an argument stands—that is, whether it is supported by sufficient reasons and not successfully undermined by counter-reasons (or whether such counter-reasons are themselves defeated). However, applying AFs to natural language remains difficult: key premises are often implicit in context, either because they are taken as obvious or because speakers prefer not to expose them to scrutiny. Moreover, while AFs are well suited for evaluating the overall acceptability of arguments in a debate, they are typically less equipped to assess the plausibility of individual premises and the strength of specific inferential links, which often depend on commonsense knowledge and pragmatic factors that escape purely formal treatments.

*Corresponding author: ppirozelli@usp.br.

The recent growth of large language models (LLMs), such as OpenAI's GPT models (Singh et al., 2025) and Meta's Llama models (Touvron et al., 2023), represents a promising development for computational argumentation. These models capture lexical, syntactic, and pragmatic regularities, and can uncover implicit connections that are difficult to formalize. Nonetheless, while LLMs can be effective in downstream tasks such as argumentative span detection and relation classification, they do not by themselves provide a systematic structure for representing and evaluating arguments. In particular, these tasks remain underspecified in terms of what should count as an argument and which criteria should guide its evaluation. Thus, off-the-shelf generations from LLMs are not a substitute for structured representations: without an explicit schema and controllable intermediate steps, outputs can be inconsistent and difficult to compare or aggregate across documents and datasets.

Hence, both AFs and LLMs offer valuable strengths as well as shortcoming. AFs provide explicit and well-defined representational structures, while LLMs offer rich contextual and implicit knowledge derived from large-scale language data; at the same time, AFs methods often struggle with the complexity and variability of real-life arguments, whereas LLM-based approaches may lack systematic and theoretically grounded standards for argument reconstruction. To bridge these perspectives, we turn to a third tradition from philosophy and argumentation theory: Informal Logic (IL). Informal Logic seeks to adapt the rigor of formal logic to the complexities of everyday reasoning. As Johnson and Blair define it, IL is the "branch of logic whose task is to develop non-formal standards, criteria, and procedures for the analysis, interpretation, evaluation, critique and construction of argumentation in everyday discourse" (Johnson & Blair, 1977, p. 158).

In this paper, we combine these approaches into a unified framework for reconstructing arguments from natural language text. We employ a multi-stage LLM-based pipeline grounded in established IL analysis, which identifies relevant argumentative components and infers their relationships. To ensure well-formed outputs, we incorporate preprocessing steps that enforce a consistent graph-structured representation.

Our main contributions are:

- A multi-stage LLM-based pipeline for argument reconstruction grounded in IL;
- An annotated IL argumentation dataset.[1]

2 Related Work

Research on argument mining has developed along two main directions: (i) formal models of argumentative interaction and (ii) methods for extracting arguments from natural language text. Formal models, particularly *argumentation frameworks*, provide abstract representations and semantics for reasoning about conflicts between arguments. *Argument mining* focuses on identifying argumentative components and relations in raw text. These strands are often pursued independently, leaving a gap between formal representations and natural language processing approaches, as well as their connection to normative evaluation standards.

2.1 Argumentation Framework

Argumentation frameworks provide an abstract, graph-based model in which arguments are represented as nodes and conflicts between them as a binary attack relation. Semantics then determine which sets of arguments can be jointly accepted (e.g., grounded, preferred, and stable extensions) (Dung, 1995; Baroni et al., 2011). Several extensions increase expressivity, including value-based AFs, bipolar AFs, and abstract dialectical frameworks (Bench-Capon, 2003; Cayrol & Lagasquie-Schiex, 2005; Brewka et al., 2010). Structured formalisms such as ASPIC+ and assumption-based argumentation reconnect abstract arguments to premises and inference rules, enabling explanations in terms of different kinds of defeats (Modgil & Prakken, 2014; Dung et al., 2009).

Despite their success, AF approaches typically assume that arguments and relations are already given, leaving their extraction from raw text outside the framework. Moreover, evaluating the plausibility of individual premises or inferential steps often requires commonsense and pragmatic knowl-

[1] Code and data are available at `https://github.com/do-ald533/llm_argumentation`.

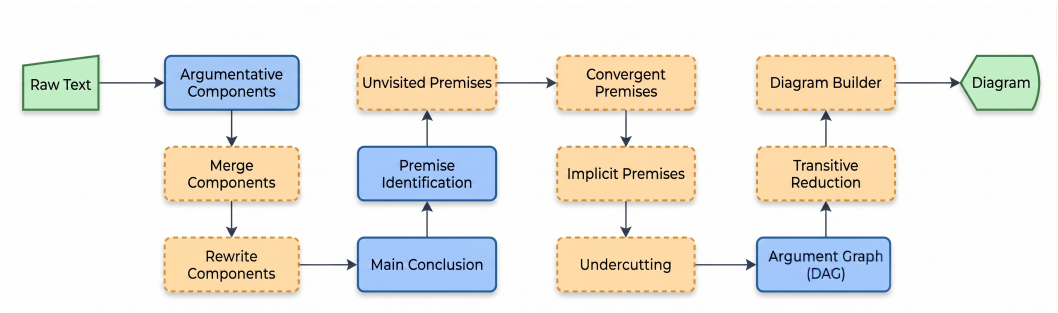


Figure 1: Overview of the system pipeline. The model converts natural language text into an argumentative directed acyclic graph. Blue boxes denote mandatory steps, while beige boxes denote optional steps.

edge that formal semantics alone do not capture. Our system addresses these limitations by using LLM-guided argument mining to construct AFs directly from natural language.

2.2 Argument Mining

Argument mining seeks to recover argumentative structure from natural language—typically by segmenting text into argumentative discourse units, identifying their roles (e.g., major claim, claim, premise), and predicting the relations of support or attack that connect them. Early work in AM relied on feature-rich statistical models with handcrafted features. Later neural approaches replaced these with pre-trained encoders, leading to substantial gains in generalization and robustness (Lippi & Torroni, 2016; Lawrence & Reed, 2020).

More recently, LLMs have been increasingly employed across a range of AM tasks (Li et al., 2025; Chen et al., 2024). LLMs have been used to assess argument quality and emotional appeal (Chen & Eger, 2025), perform stance detection through reasoning chains (Ma et al., 2024), assist in dataset creation and summarization (Li et al., 2024; Liu et al., 2023), and automate evaluation of argumentative outputs (Dhole et al., 2025). These developments mark a shift from traditional supervised pipelines toward hybrid methods combining LLM reasoning with structured representations. Our work follows this direction by integrating LLM-based argument extraction into formal argumentation frameworks, linking natural language text to abstract argumentative graphs.

3 LLM-Based Argument Mining

Despite their impressive fluency, current LLMs still struggle to perform argument mining in an end-to-end, single-pass setting. This challenge is twofold: (i) LLMs often fail to maintain long-range structure and global consistency, and (ii) argument mining remains underspecified, with no clear consensus on what constitutes an argument, how its components should be represented, or which graph structures are admissible. To address these challenges, we design a multi-stage pipeline that incrementally constructs an argumentative graph from raw text. Our framework is grounded in standard IL approaches (Walton, 1996; Freeman, 2011; Kelley, 2014; Walton et al., 2008; Walton, 2008), incorporating not only basic components such as premises and conclusions and relations such as support and attack, but also more nuanced phenomena studied in this tradition, such as implicit premises, linked premises, and undercuts.

The pipeline combines LLM-driven extraction and analysis modules, coupled with an explicit representation schema. Each stage is implemented through targeted prompts to the LLM. Its modular design allows individual components to be independently disabled or replaced with alternative implementations. The following sections describe each stage of the pipeline, with optional steps marked by an *asterisk* (*). Figure 1 shows the full pipeline.

[1] Argumentative Components. The first step consists of identifying and enumerating the argumentative components within a text. The objective is to distinguish elements that contribute to

the argument—such as claims and premises—from those that do not. The components identified at this stage constitute the foundational building blocks of the argument, with irrelevant content filtered out. Crucially, the role of a given textual span depends not only on its intrinsic content but also on its surrounding context (Opitz & Frank, 2019). This context sensitivity provides LLMs with a distinct advantage: unlike rule-based and traditional machine learning approaches, which typically rely on surface-level features, LLMs can infer whether a span contributes to an argument by leveraging the rich contextual and discourse-level information acquired during pretraining.

[2] Merge Components.* After identifying the individual argumentative components, the next step is to detect and merge those that are redundant or closely connected within the same line of reasoning. Merging involves analyzing the semantic and logical relationships among components to identify overlaps or dependencies. Components should be merged if they paraphrase one another, elaborate on the same point, or form a coherent reasoning chain—such as a conditional statement and its implication, or a component that is distributed across multiple sentences. This process ensures that the argumentation structure remains concise and logically unified, preventing fragmentation and enhancing interpretability. This step is particularly useful for longer arguments, but may be omitted for shorter ones or when a more fine-grained analysis is desired.

[3] Component Rewriting.* Once the individual components have been identified, they are rewritten for clarity. This process resolves incomplete or context-dependent expressions—such as pronouns or ellipses—by making each component self-contained and unambiguous.[2] This step primarily improves readability and may be omitted when such refinement is not required.

[4] Conclusion Identification. After extracting the argumentative components, the next step is to reconstruct the logical relations among them. As an initial step, the main conclusion is identified from the set of components.

[5] Premise Relation. Given the main conclusion, support and attack relations between argumentative components are identified using a *recursive* strategy. Specifically, the premises of each conclusion are determined, beginning with the main conclusion and progressively expanding to its supporting or attacking components. This decomposition reduces the complexity of the graph construction task, as only the premises for a single target conclusion need to be determined at a time, rather than inferring the entire structure at once. At each step, the system considers the full text, the list of argumentative components, and the current target conclusion, and returns the components that support or attack the target, or outputs `0` if no such components exist. Each component is expanded as a conclusion at most once. When a premise is identified for a given conclusion, it is added to a queue of components to be processed, following a breadth-first traversal of the argument graph. To ensure that the resulting structure is acyclic, candidate premises are restricted to unvisited components only. This prevents edges from pointing to ancestors or to components at the same hierarchical level in the partially constructed graph, thereby enforcing a directed acyclic graph structure.

[6] Check Unvisited Premises.* Because premise identification is performed locally for each target conclusion, some argumentative components may remain unassigned as premises and thus disconnected from the constructed argument graph. To address this, each unvisited component is examined to determine where it should be attached. Since unvisited components may also relate to one another, previously disconnected components can be assigned as mutual premises. When a cycle is detected, the components involved are merged into a single composite unit, and the attachment procedure is repeated for the resulting component. Although this step is optional, omitting it typically leads to missing components and a more fragmented graph.

[7] Linked and Convergent Premises.* Following IL representations, relations between premises at the same hierarchical level are classified as either linked or convergent (Walton, 1996). In a convergent relation, each premise provides an independent reason to support or attack the conclusion; in a linked relation, multiple premises jointly justify or refute the conclusion. Linked premises are

[2] In principle, this step could also be extended to summarize components that are overly long or complex, resulting in a more concise representation of the argument.

represented through an intermediate empty node, capturing their joint contribution as a single inferential unit, rather than as independent supports. This step yields a more accurate representation of the inferential structure, albeit at the cost of an increased number of nodes in the resulting graph.

[8] Implicit Premises.* Arguments often omit premises that are necessary to fully support or attack a conclusion; such arguments are known as *enthymemes* (Walton, 1987). These premises may be left unstated for various reasons: they may be assumed to be shared by interlocutors, regarded as common knowledge, or omitted to draw attention to more contentious points. *Implicit premises* are premises assumed in the argument but not explicitly expressed. Recovering them is essential for accurately reconstructing the structure of an argument, as they operate in tandem with explicit components to justify or refute a conclusion. As with explicit premises, we identify implicit ones recursively. For each inferential relation, we prompt the model to generate plausible implicit premises. In the case of convergent premises, this involves requesting common assumptions underlying the entire set. The newly generated components are added to the dictionary of argumentative elements, and the logical relations are updated accordingly to incorporate them. Although recovering implicit premises often clarifies inferential connections, it may also introduce trivial or self-evident statements, requiring subsequent filtering or interpretation.

[9] Rebuttal and Undercut.* In IL two types of attacking relations are often distinguished: in which a premise (or set of premises) directly challenges a component, and *undercuts*, in which a premise targets the inference to the conclusion itself (Pollock, 1987; 2001). Consider the argument: "The lawn is wet. Therefore, it rained." A rebuttal would be: "The lawn isn't wet at all," which directly challenges the truth of the premise. An undercutter would be: "The sprinkler ran overnight," which accepts the premise but blocks the inference to the conclusion by offering an alternative explanation. To operationalize this distinction, we examine each attacking relation (e.g., 4 attacks 6') and present the model with all inference links involving the attacked conclusion (e.g., 2 supports 6' or 6 supports 1). If the model indicates that an attacking relation is an undercut, we insert an empty intermediate node into the original inference so that the attack targets this node rather than the conclusion directly. This step enables the explicit representation of challenges to inferential links, albeit at the cost of increased structural complexity in the resulting graph.

[10] Argumentative Graph. An argument consists of propositions connected by logical relations. To represent this structure, we model the arguments as *graphs*. Each node represents an argumentative component, and each edge denotes a directed relation between components, indicating that the source node either justifies or refutes the target node. The relation between a premise and a conclusion is unidirectional, and the graph must be acyclic. Allowing a premise to be supported by its own conclusion would violate the requirement of independent justification. Accordingly, we represent arguments as *directed acyclic graphs* (DAGs), defined as $G = (N, E)$, where N is the set of nodes and E is the set of edges.

[11] Transitive Reduction.* A transitive reduction removes all edges that are not necessary to preserve reachability between nodes. In our context, this means eliminating direct connections between a premise and a conclusion when an indirect path already exists. The goal is to eliminate superfluous edges that clutter the graph without adding meaningful information. These redundant links are often a byproduct of the recursive construction: while the model may initially detect a connection between a premise and a conclusion, it may only later identify that the relation is mediated through intermediate steps. Transitive reduction thus helps clarify the underlying logical structure of the argument.

[12] Diagram.* A common challenge in argument evaluation is the need for a global view of the structure. Some logical relations may appear plausible in isolation but prove inadequate when considered in a broader context. Relying solely on metric-based evaluations—such as component classification—can underestimate the importance of key argumentative connections. To address this, we provide a diagram builder that enables visual inspection of the argumentative graph. This facilitates manual refinement of the pipeline and more informed prompting. More importantly, a visual representation aligns with what people intuitively expect from argument analysis. For this reason, argument diagramming is widely used in Informal Logic textbooks (Reed et al., 2007). Our

1. The teacher must approve all students.
2. Failing students would cause them to drop out of the course.
3. The college board does not want to lose tuition revenue from these students.
4. If academic merit took precedence, the situation would be different.
5. Academic merit does not take precedence at this college.
6. The director said at a departmental meeting that financial balance is the institution's absolute priority this semester.
7. If the board does not want to lose tuition revenue from these students, it will take measures to prevent those students from failing (for example, by instructing or pressuring staff to pass them).
8. The teacher is obliged or compelled to follow the board's directives/priorities, so will approve students when the board demands it.

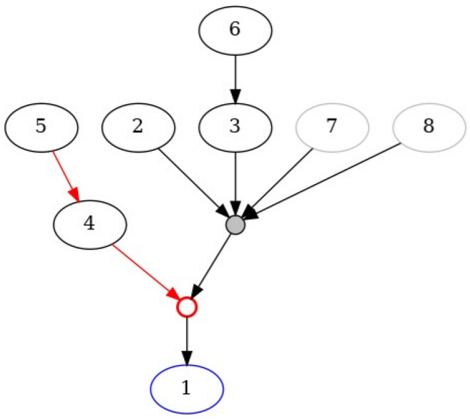


Figure 2: Diagram of the Teacher argument. Explicit premises are shown as *black nodes*, and implicit premises as *gray nodes*. *Black edges* indicate support relations, and *red edges* indicate attack relations. *Small gray nodes* represent convergent premises, i.e., premises that jointly support or attack a conclusion. *Small red nodes* represent undercutting attacks, targeting the inference rather than a specific premise. The *blue node* indicates the conclusion.

diagram builder supports visual distinctions between explicit and implicit premises, supporting and attacking relations, convergent and independent premises, and rebuttals and undercuts.

3.1 Example

We illustrate our system with an argument about the prioritization of financial interests over academic merit in a college setting extracted from (Sacrini, 2023).

Input Text

The teacher will have to approve all students, since any failure would result in the students failing to drop out of the course, and the college's board does not want to miss out on receiving any tuition fees from these students. It could be different if concern for academic merit took precedence in this college, but clearly, it is not the case, as, at the departmental meeting, the director warned that the financial balance of the institution is the absolute priority for the semester.

Figure 2 illustrates the reconstructed argument according to our full system. In the diagram, explicit premises are represented by *black nodes*, while *gray nodes* denote implicit premises. *Black edges* indicate support relations, and *red edges* indicate attack relations. *Small gray nodes* represent convergent premises—i.e., premises that work together to support or attack a conclusion. *Small red nodes* represent undercutting attacks, which target the inference itself rather than a specific premise. Finally, the *blue node* indicates the conclusion of the argument.

3.2 Data

A key challenge in evaluating argument mining methods is the limited availability of annotated datasets. Compared to other NLP tasks, resources for argumentation remain relatively scarce (Peldszus & Stede, 2015; Reed, 2006; Shnarch et al., 2020). Moreover, existing datasets differ substantially in their annotation schemes: they vary in the types of argumentative components they define (e.g., claim, premise, or more fine-grained roles such as proposal and observation) (Stab &

Gurevych, 2017; Accuosto & Saggion, 2020), in the relations they distinguish (e.g., support, attack, evidential links) (Mayer et al., 2020; Park & Cardie, 2018), and in the overall structures they assume (e.g., trees vs. graphs) (Rinott et al., 2015; Rocha et al., 2023).

This lack of consistency makes comparing models across datasets a challenging task. In particular, none of these datasets includes the exact representations we employ, such as counterarguments. They also do not indicate implicit premises or convergent arguments. To address this issue, we constructed our own dataset from *Introduction to Argumentative Analysis: Theory and Practice* by Marcus Sacrini, an IL textbook (Sacrini, 2023). The dataset contains 42 short arguments on commonsense and philosophy topics, with an average length of 332.10 characters ($\sigma = 225.20$). The diagrams in the book use a similar annotation scheme to ours and were manually annotated by the author.[3]

To ensure comparability, we selected two widely used argument mining datasets with representations relatively close to our graph-based model: AAEC Stab & Gurevych (2017) and AbstRCT Mayer et al. (2020). AAEC consists of persuasive essays written by students, whereas AbstRCT contains biomedical abstracts describing randomized controlled trials. Both datasets annotate argumentative components and relations in a way that is broadly compatible with our framework, enabling us to test our system with only minor adaptations. Both AAEC and AbstRCT include three component types—major claim, claim, and premise—as well as relations indicating support or attack. In addition, AbstRCT distinguishes between two types of attacking relations: attack and partial attack, the latter representing a weaker form of opposition. The datasets also differ in text length: arguments in AbstRCT are generally closer in size to the shorter texts in our dataset, whereas AAEC contains substantially longer documents.

For all experiments, we generated argument graphs for every instance in the dataset using three models: `GPT-4.1`, `GPT-5`, and `GPT-5-mini`. The latter offers a good compromise between output quality and cost. For each model, we generated a single set of graphs using a fixed random seed. Qualitative assessment revealed considerable variability in the generated argument structures across runs. Accordingly, generating multiple candidates and selecting the most coherent one may yield more robust results in practice.

4 Experiments

We conducted two experiments: (i) a manual evaluation of IL arguments (internal evaluation), and (ii) a quantitative evaluation on existing benchmark datasets (external evaluation).

4.1 Internal Evaluation

To assess the quality of the graphs produced by our method on the IL dataset, we adopt a three-level evaluation framework: component-, structure-, and global-level assessments, spanning from individual components to the full argumentative graph. The component level, which admits a more objective evaluation, is assessed through direct comparison between ground truth and predictions, while the structure and global levels rely on more open-ended, qualitative judgments. Detailed annotation criteria are provided in Appendix B.

4.1.1 Component Evaluation

We first evaluated our system's ability to identify basic argumentative elements—individual components, conclusions, and one-to-one relations. To this end, we compared automatically generated graphs with the ground truth diagrams. Because this evaluation was relatively straightforward, it was conducted by a single annotator. The following tasks were assessed quantitatively:

- **Span Detection.** The evaluation assessed whether the system correctly identified the text spans corresponding to argumentative components (claims and premises). Precision, recall, and F1-score were computed against the reference.
- **Conclusion Detection.** Accuracy was calculated to determine how well the system identified the main conclusions, based on the annotated conclusions in the reference.

[3] They also included additional features, such as logical type support, which we did not incorporate.

Table 1: Evaluation results for the generated argument graphs. Component evaluations report precision, recall, F1-score, and accuracy, whereas structure and global evaluations report the mean score (1–3 scale) and exact agreement between annotators.

Task / Criterion	Precision	Recall	F1	Accuracy	Quality (*EA*)
Component Evaluation					
Span Detection	80.31	72.87	75.70		
Conclusion Detection				92.50	
Relation Detection				80.57	
Structure Evaluation					
Implicit Premises					2.86 (0.95)
Undercuts					2.90 (0.88)
Convergent/Linked Premises					2.85 (0.80)
Global Evaluation					
Completeness					2.96 (0.97)
Faithfulness					2.92 (0.88)

- **Relation Detection.** Accuracy was used to assess whether directed links between components were correctly identified. Cases with fewer than two shared components between reference and generated graphs were excluded. The evaluation focused on premise–conclusion links, disregarding premise relations (e.g., convergence, independence), counterarguments, implicit premises, and support types.

The upper block of Table 1 reports the outcomes of the comparison between automatically generated graphs and the gold-standard diagrams. Minor wording differences—such as changes in tense, modal verbs, or voice—were intentionally ignored during evaluation, as they do not affect the argumentative content.

Discussion Results showed near-perfect accuracy (92.50%) in conclusion detection. Occasional errors occurred when the system merged a premise and a conclusion or when the conclusion was implicit, something the system could not handle. For span detection, scores were lower (precision 80.31%, recall 72.87%, F1-score 75.70%), likely due to annotation inconsistencies. The system often merged sentences that the gold standard had split into two, although both expressed the same meaning. In such cases, multiple valid annotations were possible, so these discrepancies did not necessarily indicate model error. For relation detection, the evaluation considered only arguments containing at least two matching components. Within this constrained setup, the generated and reference graphs aligned closely, achieving an accuracy of 80.57%.

4.1.2 Structure Evaluation

A second set of evaluations focused on the structural patterns of the arguments—that is, whether the components were correctly organized. In particular, the analysis examined the appropriate use of implicit premises, undercuts, and the structural organization of premises (convergent or linked). Annotators had access to the original text, and reference diagrams were used as a point of comparison, although alternative interpretations were accepted. To assess these structural aspects, two independent annotators were employed. The evaluation criteria were as follows:

- **Implicit Premises.** For each inferential relation, annotators assessed whether implicit premises were adequate, missing, or redundant.
- **Undercuts.** Annotators evaluated whether undercuts were correctly identified.
- **Convergent and Linked Premises.** Annotators judged whether the premises were appropriately classified as linked (joint support) or convergent (independent support).

Appendix B provides a detailed description of the annotation criteria for this and the subsequent evaluation.

Discussion The middle block of Table 1 presents the results of these evaluations. Each criterion was rated on a 1–3 scale, and both the mean score across annotators and their rate of exact agreement

(EA) were reported.[4] Overall, the arguments were rated as very good across all criteria, with a high level of agreement among annotators: Implicit Premises (M = 2.86, EA = 0.95), Undercuts (M = 2.90, EA = 0.88), and Convergent/Linked Premises (M = 2.85, EA = 0.80). The evaluation of undercuts, however, was less reliable, as only eight arguments included them. In these cases, much of the reasoning relied on counterfactual judgments rather than explicit textual evidence.

4.1.3 GLOBAL EVALUATION

Finally, a set of global evaluations was conducted to capture overall properties of the generated argument graphs. These criteria provided a holistic assessment of whether the graphs adequately covered the main argumentative content while remaining faithful to the original text. As in the previous evaluation, two annotators independently performed the assessments.

- **Completeness.** Assessed whether the graph captured all major claims and premises expressed in the text.
- **Faithfulness.** Assessed whether the graph avoided introducing information not supported by the text (i.e., no hallucinated nodes or edges).

Discussion The bottom block of Table 1 summarized the evaluations of completeness and faithfulness. Both criteria were rated on a 1–3 scale, and the mean score and exact agreement between annotators were reported. Overall, the results were very good for both Completeness (M = 2.96, EA = 0.97) and Faithfulness (M = 2.92, EA = 0.88). These scores indicated that the generated graphs generally captured the main argumentative structures of the texts while maintaining high fidelity to their content. The strong agreement between annotators further suggests that the overall quality of the generated argument graphs was consistent and reliable.

4.2 EXTERNAL EVALUATION

To analyze the quality of our LLM-based framework on public datasets, we employ the simplest representation, as to permit comparison with those different formats. Thus, we turn off the steps indicated by an asterisk in our pipeline — i.e., steps 1, 4, 5, 6.[5] Also, as mentioned above, both AAEC and AbstRCT classify components into three types: major claim, claim, and premise, whereas we only use conclusion and premise. Thus, we map components to these categories based on the structure of the generated graphs: the root node is considered the major claim, components directly attached to it are treated as claims, and all remaining lower-level elements are considered premises. As for the partial-attack relation from the AbstRCT dataset, we developed a special prompt to detect this as a third type of relation.

We evaluate the generated argument structures using standard argument mining tasks commonly adopted in the literature (Morio et al., 2022). Although our framework produces complete argument graphs, breaking the evaluation down into these tasks enables direct comparison with existing systems. Task definitions and evaluation protocols are detailed below:

- **Span Identification.** This task consists of detecting portions of the input text corresponding to argumentative components. A component is considered correctly identified if it exactly matches a ground-truth component. Performance is evaluated using the (micro) *F1-score* over the predicted and ground-truth spans.
- **Component Classification.** This task assigns a component type to each span according to the datasets' annotation scheme (premise, claim, or major claim). Performance is evaluated using *F1* and *Macro-F1* scores over the component labels.
- **Relation Classification.** This task determines the argumentative relations between pairs of components and involves two aspects. First, the model predicts whether a relation exists between two components, regardless of its type. This evaluates the argument graph structure and is reported as *Link* using *F1*. Second, the model assigns a label to each detected

[4] Cohen's kappa was not reported, as it tends to be unreliable when score distributions are highly skewed.

[5] We do not enforce the DAG constraint and therefore disable step 10 either, as doing so would require merging cyclic components, which in turn would entail modifying the original components.

Table 2: Results on the AbstRCT and AAEC datasets. The full pipeline includes the argument component identification step, whereas the gold setting uses ground-truth components.

Dataset / Model	**Span**	**Component**		**Relation**		
	F1	**F1**	**Macro**	**Link**	**F1**	**Macro**
AbstRCT						
LLM-based pipeline (full)	59.65	27.64	19.26	20.82	19.33	8.8
Morio et al. (2022)	**70.93**	**64.78**	**44.56**	39.74	38.71	**33.94**
LLM-based pipeline (gold comp.)	-	61.93	38.53	**51.51**	**46.21**	23.45
AAEC						
LLM-based pipeline (full)	33.03	22.70	13.26	0.41	0.34	0.19
Morio et al. (2022)	**85.20**	**75.66**	**67.03**	**55.72**	**55.17**	**41.92**
LLM-based pipeline (gold comp.)	-	62.64	54.15	22.88	22.67	23.11

relation (e.g., support or attack). Relation type classification is evaluated using *F1* and *Macro-F1* scores.

Table 2 reports the results of this experiment. The full pipeline (first row) yields poor overall performance. This is largely because the model is not designed to select argument spans under strict criteria, resulting in low overlap with the ground truth annotations. Since the evaluation requires exact span matching, even minor deviations lead to substantial performance drops in all tasks, as errors propagate (see Appendix C for a detailed analysis).

The third row shows the results obtained when gold-standard components are provided to the system. Performance improves substantially, indicating that the main bottleneck lies in span detection rather than in component or relation classification. Under these conditions, our approach achieves its best results on AbstRCT, reaching 51.51 in link prediction and 46.21 in relation detection—improvements of $+11.77$ and $+7.5$ over the state-of-the-art baseline. This suggests that the model effectively captures relational structure once the components are known. Performance is also stronger on AbstRCT than on AAEC, likely due to its shorter, more controlled texts, which resemble canonical IL examples.

5 Conclusion

Argumentation frameworks and LLMs exhibit important limitations in argument mining: methods based on argumentation frameworks often struggle with the complexity and variability of real-world arguments, whereas LLM-based approaches lack systematic and theoretically grounded standards for argument reconstruction. In this paper, we proposed an LLM-based framework grounded in Informal Logic, designed to reconstruct argumentative structures from natural language text. The system leverages LLMs' commonsense knowledge to recover arguments while imposing a standard graph-like representation. Our results show that the system can recover a range of Informal Logic–style arguments and, when adapted to different annotation schemes, achieves reasonable performance across benchmark datasets. At the same time, the results highlight limitations in component identification and the need for improved prompting strategies. Future work will focus on evaluating the system on more complex and realistic datasets, refining the framework to capture richer argumentative phenomena (e.g., types of premises and argument schemes), and incorporating Informal Logic criteria for evaluation.

Acknowledgment

The authors of this work would like to thank the Center for Artificial Intelligence (C4AI-USP) and the support from the São Paulo Research Foundation (FAPESP grant #2019/07665-4) and from the IBM Corporation. F. G. C. was partially supported by CNPq grants 312180/2018-7 and 305753/2022-3. The authors also thank support by CAPES - Finance Code 001.

REFERENCES

Pablo Accuosto and Horacio Saggion. Mining arguments in scientific abstracts with discourse-level embeddings. *Data & Knowledge Engineering*, 129:101840, 2020.

Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. Semantics of abstract argument systems. *Handbook of Formal Argumentation*, 1:159–236, 2011.

Trevor J. M. Bench-Capon. Value-based argumentation frameworks. In *Proceedings of the 9th International Workshop on Non-Monotonic Reasoning*, pp. 443–454, 2003.

Gerhard Brewka, Sylwia Polberg, and Stefan Woltran. Abstract dialectical frameworks. *Proceedings of the Twelfth International Conference on Principles of Knowledge Representation and Reasoning (KR 2010)*, pp. 102–111, 2010.

Claudette Cayrol and Marie-Christine Lagasquie-Schiex. On the acceptability of arguments in bipolar argumentation frameworks. *Proceedings of the Eighth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2005)*, pp. 378–389, 2005.

Guizhen Chen, Liying Cheng, Luu Anh Tuan, and Lidong Bing. Exploring the potential of large language models in computational argumentation, 2024.

Yanran Chen and Steffen Eger. Do emotions really affect argument convincingness? a dynamic approach with LLM-based manipulation checks. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 24357–24381, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5.

Kaustubh Dhole, Kai Shu, and Eugene Agichtein. ConQRet: A new benchmark for fine-grained automatic evaluation of retrieval augmented computational argumentation. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5687–5713, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6.

Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, 1995.

Phan Minh Dung, Robert A Kowalski, and Francesca Toni. Assumption-based argumentation. *Artificial Intelligence*, 171(10-15):715–741, 2009.

Frans H van Eemeren. *Argumentation theory: A pragma-dialectical perspective*. Springer, 2018.

James B. Freeman. *Argument Structure: Representation and Theory*. Springer, New York, 2011.

Michael HG Hoffmann and Richard Catrambone. Bad arguments and objectively bad arguments. *Informal Logic*, 43(1):23–90, 2023.

Ralph Henry Johnson and J. Anthony Blair. *Logical Self-Defense*. Mcgraw-Hill, Toronto, Canada, 1977.

David Kelley. *The Art of Reasoning: An Introduction to Logic and Critical Thinking*. W. W. Norton & Company, London, 2014.

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

Hao Li, Yuping Wu, Viktor Schlegel, Riza Batista-Navarro, Tharindu Madusanka, Iqra Zahid, Jiayan Zeng, Xiaochi Wang, Xinran He, Yizhi Li, and Goran Nenadic. Which side are you on? a multi-task dataset for end-to-end argument summarisation and evaluation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 133–150, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

Hao Li, Viktor Schlegel, Yizheng Sun, Riza Batista-Navarro, and Goran Nenadic. Large language models in argument mining: A survey. *arXiv preprint arXiv:2506.16383*, 2025.

Marco Lippi and Paolo Torroni. Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology (TOIT)*, 16(2):1–25, 2016.

Yikang Liu, Ziyin Zhang, Wanyang Zhang, Shisen Yue, Xiaojing Zhao, Xinyuan Cheng, Yiwen Zhang, and Hai Hu. Argugpt: evaluating, understanding and identifying argumentative essays generated by gpt models, 2023.

Junxia Ma, Changjiang Wang, Hanwen Xing, Dongming Zhao, and Yazhou Zhang. Chain of stance: Stance detection with large language models, 2024.

Tobias Mayer, Elena Cabrio, and Serena Villata. Transformer-based argument mining for healthcare applications. In *ECAI 2020*, pp. 2108–2115. IOS Press, 2020.

Sanjay Modgil and Henry Prakken. The aspic+ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1):31–62, 2014.

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

Juri Opitz and Anette Frank. Dissecting content and context in argumentative relation analysis. *arXiv preprint arXiv:1906.03338*, 2019.

Joonsuk Park and Claire Cardie. 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)*, 2018.

Andreas Peldszus and Manfred Stede. An annotated corpus of argumentative microtexts. In *Argumentation and Reasoned Action: Proceedings of the 1st European Conference on Argumentation, Lisbon*, volume 2, pp. 801–815, 2015.

John L Pollock. Defeasible reasoning. *Cognitive science*, 11(4):481–518, 1987.

John L Pollock. Defeasible reasoning with variable degrees of justification. *Artificial intelligence*, 133(1-2):233–282, 2001.

Chris Reed. Preliminary results from an argument corpus. *Linguistics in the twenty-first century*, pp. 185–196, 2006.

Chris Reed, Douglas Walton, and Fabrizio Macagno. Argument diagramming in logic, law and artificial intelligence. *The Knowledge Engineering Review*, 22(1):87–109, 2007.

Ruty Rinott, Lena Dankin, Carlos Alzate, Mitesh M Khapra, Ehud Aharoni, and Noam Slonim. Show me your evidence-an automatic method for context dependent evidence detection. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 440–450, 2015.

Victor Hugo Nascimento Rocha, Igor Cataneo Silveira, Paulo Pirozelli, Denis Deratani Mauá, and Fabio Gagliardi Cozman. Assessing good, bad and ugly arguments generated by chatgpt: a new dataset, its methodology and associated tasks. In *EPIA Conference on Artificial Intelligence*, pp. 428–440. Springer, 2023.

Marcus Sacrini. *Introdução à Análise Argumentativa: Teoria e Prática*. Paulus Editora, São Paulo, 2 edition, 2023.

Eyal Shnarch, Leshem Choshen, Guy Moshkowich, Noam Slonim, and Ranit Aharonov. Unsupervised expressive rules provide explainability and assist human experts grasping new domains. *arXiv preprint arXiv:2010.09459*, 2020.

Aaditya Singh, Adam Fry, Adam Perelman, et al. Openai gpt-5 system card, 2025. URL `https://arxiv.org/abs/2601.03267`.

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

Hugo Touvron, Louis Martin, Kevin Stone, et al. Llama 2: Open foundation and fine-tuned chat models, 2023.

Douglas Walton. *Informal Fallacies: Towards a Theory of Argument Criticisms*. Companion series. J. Benjamins Publishing Company, 1987. ISBN 9789027250056.

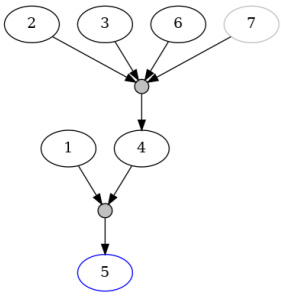
Douglas Walton. *Argument structure: A pragmatic theory*. University of Toronto Press Toronto, 1996.

Douglas Walton. *Informal Logic: A Pragmatic Approach*. Cambridge University Press, 2 edition, 2008.

Douglas Walton, Christopher Reed, and Fabrizio Macagno. *Argumentation schemes*. Cambridge University Press, 2008.

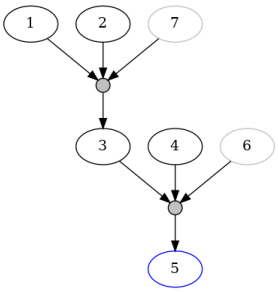
A DIAGRAMS

We present additional examples of argument graphs from our dataset. Nodes represent argumentative components: black nodes indicate explicit premises, while gray nodes denote implicit premises. Edges encode relations, with black edges representing support and red edges indicating attacks. Smaller gray nodes correspond to convergent premises. Undercutting relations—i.e., attacks on inferences—are shown as red edges terminating in red nodes. The conclusion is highlighted in blue.



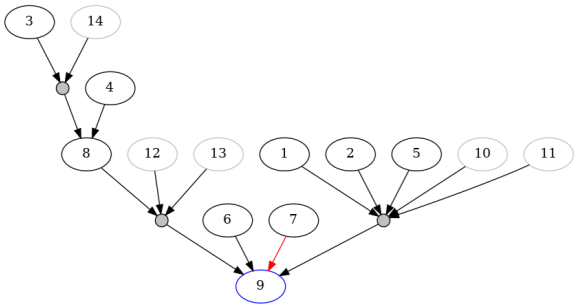
1 Nothing is demonstrable unless its negation implies a contradiction.
2 Nothing that is distinctly conceivable implies a contradiction.
3 Anything we conceive as existing can also be conceived as not existing.
4 No being's non-existence implies a contradiction.
5 Consequently, no being's existence is demonstrable.
6 Whenever we can conceive a being's non-existence, that conception counts as a distinct conception (i.e., conceiving a being as non-existent makes its non-existence distinctly conceivable).
7 For every being, we can conceive it as existing (so premise 3 applies to any being).

Figure 3: Text: "Nothing is demonstrable, unless the contrary implies a contradiction. Nothing, that is distinctly conceivable, implies a contradiction. Whatever we conceive as existent, we can also conceive as non-existent. There is no being, therefore, whose non-existence implies a contradiction. Consequently there is no being, whose existence is demonstrable. I propose this argument as entirely decisive, and am willing to rest the whole controversy upon it."



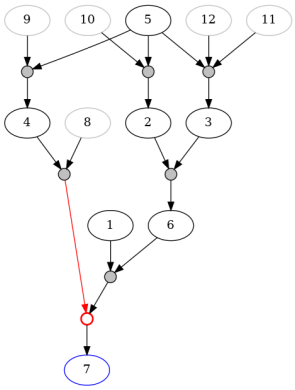
1 Everything that comes into existence has causes different from itself.
2 The universe came into existence.
3 Therefore, the universe has causes different from itself.
4 Nothing we can conceive is different from the universe, because anything we conceive is part of it.
5 Thus, the causes of the universe are completely unknowable.
6 If something cannot be conceived by us, then it is completely unknowable to us.
7 If something cannot be conceived by us, then it is completely unknowable to us.

Figure 4: Text: "Everything that comes into existence has causes different from itself. The universe came into existence. Therefore, it has causes different from itself. However, nothing we can conceive of is different from the universe, for everything we can conceive of as something is already part of the universe. Thus, the causes of the universe are completely unknowable."



1 Beavers build very complex dams that create large lakes.
2 Beaver dams have the same concave shape as human dams, a form that resists water force.
3 Humans learned to build dams only with great difficulty.
4 Humans acquired dam-building through a vast accumulation of cultural knowledge developed over centuries.
5 Beavers do not possess such accumulated cultural knowledge.
6 Beavers are animals and lack complex cultural knowledge built up over centuries.
7 Beavers instinctively know how to build dams correctly.
8 It is unclear how beavers could be guided to construct such complex dams.
9 The Lord God inspires and enables beavers to perform works beyond their limited animal capacities.
10 Constructing dams with such complex, water-resistant concave designs cannot be produced by animals lacking accumulated cultural knowledge through ordinary natural mechanisms alone.
11 If an animal's complex constructions cannot be accounted for by natural mechanisms or cultural learning, then they are explained by divine inspiration from the Lord God.
12 If a complex behavior of animals cannot be explained by natural or cultural causes, then the correct explanation is divine inspiration by the Lord God.
13 The Lord God exists and is capable of inspiring animals to perform actions beyond their natural capacities.
14 If humans learn a technology only with great difficulty, then mastery of that technology depends on a vast, cumulative body of cultural knowledge developed and transmitted across many generations.

Figure 5: Text: "Beavers build very complex dams that create large lakes. These dams are built in the same concave shape as those constructed by humans — one of the shapes that best resists the force of water. However, humans learned to build dams only with great difficulty. They owe this ability to a vast accumulation of cultural knowledge developed and consolidated over centuries. This is not the case with beavers. They are animals that do not possess complex cultural knowledge built up over centuries. Beavers simply know how to build dams correctly. How, then, could they be guided to perform constructions of such complexity? Clearly, it is the Lord God who inspires them and allows them to carry out works so far beyond their limited animal capacities."



1 The thieves fled and had only two possible routes - left via the long corridor or right via the short corridor.
2 If they had taken the longer corridor, the guard there would have seen them.
3 The guard saw nothing.
4 The guard might have been sleeping, but that is highly unlikely.
5 I know him personally; he is a serious, well-trained professional.
6 Therefore, they certainly did not take the longer corridor.
7 Thus, they must have escaped through the shorter corridor.
8 If the guard was very unlikely to be sleeping, then his not having seen anything is strong evidence that the thieves did not take the longer corridor.
9 A serious, well-trained professional guard would not be asleep or inattentive on duty and therefore would have detected the thieves if they had passed along the long corridor.
10 A serious, well-trained professional guard would not have been sleeping on duty and would have been attentive enough to have seen the thieves if they had taken the longer corridor.
11 Serious, well-trained guards do not sleep on duty and are very unlikely to be negligent.
12 A serious, well-trained guard who is awake and attentive would have noticed anyone passing through the long corridor.

Figure 6: Text: “The thieves fled and there are only two paths they could have taken — to the left, through the long corridor, or to the right, through the short corridor. If they had taken the longer corridor, they would have been seen by the guard who was there. But the guard saw nothing. It might be possible that the guard was sleeping, but that is highly unlikely. I know him personally, I know he is a serious and well-trained professional. Therefore, it is certain they did not take the longer corridor. Thus, they must have escaped through the shorter corridor.”

B ANNOTATION CRITERIA

In this section, we describe how we organized the methodology for annotation across the different evaluation stages.

For the first evaluation, referred to as Component Evaluation, we enlisted a colleague with experience in argumentation mining. The annotator assessed whether each pair of components—ground truth and prediction—referred to the same content, disregarding minor differences in wording.

For the subsequent evaluations—Structure and Global—we employed two independent evaluators. We began by presenting the evaluation criteria and explaining in detail how each task should be performed. Then, we provided five representative argumentative examples, which had been personally annotated by the authors, so that the evaluators could apply the criteria in practice. After completing this initial annotation exercise, the evaluators returned their annotations and shared questions and feedback, allowing us to identify ambiguities and refine the guidelines. Once the methodology was finalized, the evaluators proceeded to annotate the complete set of arguments following the revised instructions.

Evaluators followed a set of well-defined criteria, each assessed on a three-point scale (1–3). They had access to the original text, as well as to the reference diagrams. They were instructed to use the latter as a point of comparison, with the understanding that alternative interpretations were possible. The criteria were designed to capture both the internal quality of the argument structure and its faithfulness to the original text. Table 3 summarizes these criteria and their corresponding scales.

Criterion	Scale
Implicit Premises	
Evaluates whether the argument includes all necessary premises and whether they are well justified.	1 = crucial premises missing 2 = redundant or vague 3 = complete and well justified
Counter-Arguments	
Assesses the correctness of counter-arguments in the argument.	1 = incorrect or absent 2 = partially correct (e.g., should be a direct attack rather than a counter-argument) 3 = correct and complete (or not required)
Convergent/Divergent Premises	
Verifies whether convergent and divergent premises are correctly identified according to the text.	1 = incorrect 2 = partially correct 3 = correct in relation to the text
Completeness	
Measures the extent to which the argument includes all central claims and premises.	1 = important parts missing 2 = partially complete 3 = covers all main claims and core premises
Fidelity	
Evaluates the accuracy of the argument's content in relation to the source text.	1 = contains information not present in or distorts the original text 2 = generally faithful but with minor additions or modifications 3 = fully faithful to the text

Table 3: Criteria used for the Structure and Global Evaluations.

C Analysis of Component Similarity and Its Impact on Performance

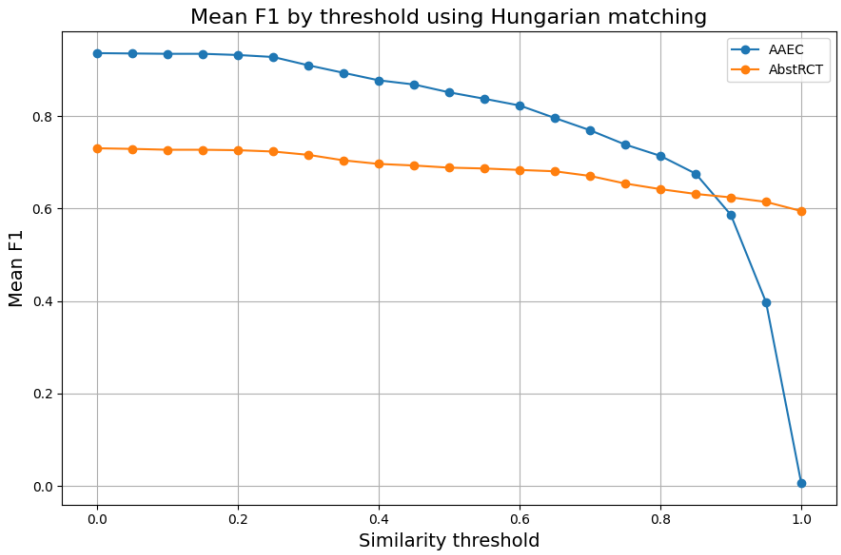


Figure 7: Mean F1-score as a function of the similarity threshold between predicted and gold components, using Hungarian matching.

To investigate the performance degradation observed in Table 2, we analyze similarity between predicted and gold components independently of exact span matching. Figure 7 reports the mean F1-score as a function of a similarity threshold, using optimal (Hungarian) matching and character-level overlap.

At moderate thresholds (e.g., 0.6), performance remains relatively high (around 0.82 for AAEC and 0.68 for AbstRCT), indicating substantial lexical overlap between predictions and gold annotations. Even at a stricter threshold (0.8), F1-scores remain above 0.60, suggesting that predicted components are often closely aligned in surface form.

However, performance drops sharply at higher thresholds. For AAEC, the F1-score declines rapidly beyond 0.9, approaching zero at 1.0, indicating that exact matches are rare despite high similarity. A similar, though less pronounced, trend is observed for AbstRCT. This highlights a limitation of strict evaluation protocols, where small boundary deviations invalidate otherwise strong matches.

These results support the hypothesis that component identification is the main bottleneck. The model frequently produces spans that are highly similar to the gold annotations but fail to meet exact matching criteria, likely due to the specificity of annotation guidelines and the absence of explicit constraints in the prompting strategy.

This analysis also explains the gap between the full pipeline and the gold-component setting in Table 2: when exact spans are provided, relation prediction improves significantly. Thus, the issue lies not in understanding argumentative content, but in aligning generated spans with annotation standards.

Overall, improving component identification—via better prompting, post-processing, or alignment—offers the greatest potential for enhancing end-to-end performance.