

A Multi-Agent LLM Pipeline for Legal Argument Mining

Safoura Banihashemi
safoura.banihashemi@studio.unibo.it
University of Bologna
Bologna, Italy

Giulia Grundler[✉]
giulia.grundler2@unibo.it
DISI - University of Bologna
Bologna, Italy

Andrea Galassi
a.galassi@unibo.it
DISI - University of Bologna
Bologna, Italy

Abstract

Accurate extraction of argumentative information from legal documents allows transforming legal documents into structured representations that can support downstream applications, such as information retrieval, document analysis, and legal decision support. Legal Argument Mining addresses the identification and classification of argumentative structures and their relations in legal texts. Such a task can be decomposed into multiple interdependent reasoning steps. In this work, we investigate the implementation of an argument mining pipeline using LLM agents, each one dedicated to a separate task. We evaluate our system under three different settings that progressively expose the impact of error propagation, ranging from independent evaluation of each agent to a realistic end-to-end scenario. We perform an experimental evaluation on the Demosthenes corpus, comparing an open-weight model against a commercial one. Results show that cascading errors substantially degrade downstream performance, making the first stages of the pipeline a significant bottleneck. Our findings highlight the importance of evaluating modular LLM systems in their end-to-end behavior in addition to individual task performance.

CCS Concepts

• Computing methodologies → Natural language processing; Information extraction; • Applied computing → Law.

Keywords

Legal Argument Mining, Large Language Models, European Court of Justice

ACM Reference Format:

Safoura Banihashemi, Giulia Grundler, and Andrea Galassi. 2026. A Multi-Agent LLM Pipeline for Legal Argument Mining. In *Proceedings of the 2026 ACM Symposium on Document Engineering (DocEng '26)*, August 25–28, 2026, Fribourg, Switzerland. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3820755.3833404>

1 Introduction

Legal Argument Mining (LAM) is concerned with the automatic identification, extraction, and classification of arguments from legal texts, such as court decisions, statutes, and contracts [11]. LAM is an important document understanding task: legal professionals such as judges, lawyers, and legal researchers, analyze large collections of lengthy and complex documents to identify relevant

arguments, supporting evidence, applicable legal principles, and precedents. The automatic extraction of this information enables the transformation of legal texts into structured representations, potentially reducing manual analysis, facilitating legal research, and improving related tasks such as document retrieval, summarization, and legal outcome prediction by making argumentative content more accessible and explainable [8, 17].

Compared to general argument mining, the legal domain presents unique challenges due to its domain-specific language, and suffers from a scarcity of public datasets. While Large Language Models (LLMs) offer a potential solution through zero/few-shot capabilities and long-context handling [10], recent literature shows they do not consistently outperform smaller, domain-specific models, highlighting that complex legal reasoning remains a challenge [1, 18].

Multi-agent systems (MAS) can help decompose a complex problem into smaller, specialized sub-tasks, each assigned to an autonomous agent with distinct prompts and roles. This collaborative architecture allows agents to act as specialized modules that refine, validate, or build upon the outputs of preceding agents. In the context of LLM-based systems, such architectures aim to improve reasoning, planning, and task decomposition [7]. However, since downstream agents operate on the outputs of preceding components, mistakes introduced in early stages may accumulate, propagating errors, and substantially affect overall system performance. The use of MAS systems has recently begun to emerge also in the legal domain, where reasoning is inherently hierarchical, and such frameworks can offer significant advantages over monolithic approaches [9, 13, 15, 16]. However, to the best of our knowledge, multi-agent architectures have not yet been investigated for LAM.

In this paper, we model LAM tasks as a five-stage multi-agent pipeline comprising argument detection, argument component classification, premise-type classification, argument schema classification, and relation identification. We evaluate the system under three configurations: Stand-Alone Agents, Sequential Evaluation, and End-to-End Pipeline, to quantify both the intrinsic capabilities of individual agents and the impact of error propagation. Experiments on the Demosthenes corpus show that, while modern LLMs can achieve strong performance on individual tasks, error propagation remains a limiting factor in end-to-end LAM pipelines. We make our code publicly available.¹

2 Related Work

Argument mining in the legal domain presents unique challenges due to its formal language, domain-specific terminology, complex discourse structures and reasoning patterns. A known issue in LAM research is the scarcity of publicly available datasets. Annotating legal texts requires specialized domain expertise, leading to high



This work is licensed under a Creative Commons Attribution 4.0 International License. *DocEng '26, Fribourg, Switzerland*

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2786-3/2026/08

<https://doi.org/10.1145/3820755.3833404>

¹<https://github.com/safoura-banihashemi/LegalAgent/>

Table 1: Dataset composition.

Split	Docs	Arg	Premise	Conclusion	Factual	Legal	Aut	Class	Itpr	Prec	Princ	Rule	Sup	Att
Test	10	612	570	42	385	213	6	13	63	113	3	87	559	21
Total	40	2,535	2,375	160	1,576	906	53	56	297	504	15	323	2,257	143

annotation costs and substantial time investment. Consequently, existing corpora are often limited in size, specific to single jurisdictions, or restricted to narrow legal sub-domains [5, 17].

LLMs show promising abilities to overcome these challenges, as they can achieve competitive performance in zero-shot or few-shot configurations without the need to be fine-tuned on task-specific data, and can also work with large contexts useful for long legal documents [10]. Berghegger et al. [2] utilize LLMs for argument detection on the ECHR dataset [12]. Gray et al. [4] leverage LLMs to generate case-based arguments and counterarguments following specific argument schemes. Jiang and Yang [8] use the legal syllogism to improve Legal Judgment Prediction, prompting LLMs to identify premises and conclusions, and show how explicitly structuring reasoning according to legal principles can significantly improve the accuracy and reliability of legal decision-making.

Nonetheless, LLMs do not consistently outperform smaller domain-specific encoder-based models. Al Zubaer et al. [18] evaluate GPT models for the argumentative classification task on the ECHR dataset [12], obtaining results below the Legal-BERT baseline. Similar conclusions are drawn by Alfieri et al. [1], that compare different demonstration selection strategies for few-shot prompting on the Demosthenes dataset [6].

Finally, while recent studies have shown that LLMs can be effective for relation identification tasks [3], their application to legal documents remains largely unexplored.

3 Data

We utilize the Demosthenes corpus [6, 14], which contains 40 judgments in English on Fiscal State Aid issued by the Court of Justice of the European Union (CJEU). In each document, there are one or more sections named "Findings of the court", where the court expresses the decision and illustrates its reasoning. The content of these sections has been manually annotated by legal experts, analyzing a broad range of legal reasoning patterns. The annotation follows a hierarchical schema, where the findings of the court are represented as chains of inferences that support a final conclusion.

Each sentence belonging to an argument chain is labeled as either a *Premise* or a *Conclusion*, with the conclusion being the final step of an argumentative chain. Premises are further classified as *Factual*, when detailing case events; *Legal*, when they address laws, principles, or precedents; or as both, when they contain mixed content. Legal premises are additionally annotated with one or more argument schemes: *Rule*, referring to the application of legislative provisions; *Precedent*, based on previous CJEU case law; *Authoritative*, relying on authoritative sources such as Advocate General opinions; *Classification*, defining whether a fact or entity falls under a legal concept; *Interpretative*, assigning meaning to legal sources through interpretative reasoning; and *Principle*, applying general legal principles to the case.

In addition, the corpus is annotated with the relations between the arguments that belong to the same argumentative chain. Five types of relations are defined: *Support from Premise(s)*, *Attack*, *Support from Failure*, *Undercut*, and *Rephrase*. Given the highly unbalanced distribution of the labels, in this work we consider only the two that account for more than 90% of the total relations. The *Support* relation indicates that a proposition is the logical outcome of one or more premises. It is the most frequent type of relation, used by the CJEU to directly justify its decisions on the parties' claims. Conversely, the *Attack* relation indicates that a proposition is contradicted by one or more premises. The attacked proposition is a statement by one party, mentioned by the Court to deny it.

We test our framework on 10 randomly selected documents. The composition of the dataset and our test set is reported in Table 1.

4 Method

4.1 Architecture

We build our multi-agent system using LangGraph, a graph-based orchestration framework for dynamic, multi-agent LLM systems, where computation is represented as a directed graph of interacting nodes linked by conditional transitions and shared state. It supports complex execution flows with branching, cycles, parallel execution and persistent memory, making it particularly suitable for multi-step reasoning tasks. Each agent of the system operates as an independent node in a directed execution graph. Intermediate outputs are represented as structured JSON objects, allowing controlled experimentation on single nodes and ablation studies.

The system is composed of 5 nodes, each one dedicated to a separate task.

Argument component detection (ACD). The agent receives a subsection of the document that starts with one "Findings of the court" and terminates before the next one. The agent segments it into a list of sentences, and labels each sentence as argumentative or non-argumentative.

Argument component classification (ACC). Given an argumentative sentence, the agent classifies it as premise or conclusion.

Premise Type Classification (PTC). A multi-label classification where a premise is classified as legal and/or factual.

Argument Scheme Classification (ASC). A multi-label classification where a legal premise is classified according to its scheme(s).

Relation Identification (RI). For each argumentative chain, the agent is required to determine the presence and type of a relation between each pair of components. We evaluate the agent on two dimensions: Link Detection, its ability to infer the presence or absence of a relation, and Relation Classification, the ability to classify the relation type as Support from Premise or Attack.

Table 2: Results on the ACD, ACC, PTC, and ASC tasks. The performances are grouped by their evaluation setting.

Setting / Model	ACD			ACC			PTC			ASC						
	Arg	Non	M-F ₁	Prem	Conc	M-F ₁	F	L	M-F ₁	Aut	Class	Itpr	Prec	Princ	Rule	M-F ₁
<i>Stand-Alone</i>																
Gemma (Zero-Shot)	.84	.80	.82	.95	.59	.77	.95	.85	.90	.71	.39	.59	.93	.36	.90	.65
Gemma (Detailed)	.74	.59	.67	.96	.68	.82	.94	.85	.90	.60	.07	.58	.92	.31	.92	.56
Gemini (Detailed)	.89	.85	.87	.97	.72	.84	.96	.92	.94	.57	.00	.50	.91	.27	.97	.53
<i>Sequential Evaluation</i>																
Gemma (Zero-Shot)	-	-	-	.79	.52	.65	.74	.77	.75	.50	.50	.40	.62	.67	.65	.55
Gemma (Detailed)	-	-	-	.84	.65	.75	.82	.85	.84	.67	.11	.47	.78	.00	.82	.48
Gemini (Detailed)	-	-	-	.97	.71	.84	.96	.92	.94	.53	.00	.52	.91	.31	.97	.54
<i>End-to-End Pipeline</i>																
Gemma (Zero-Shot)	-	-	-	.67	.37	.52	.40	.62	.51	.20	.40	.25	.42	.40	.39	.34
Gemma (Detailed)	-	-	-	.61	.46	.54	.41	.63	.52	.33	.09	.32	.49	.00	.46	.28
Gemini (Detailed)	-	-	-	.89	.50	.69	.81	.88	.84	.35	.00	.45	.86	.31	.81	.46

Table 3: Results for the Relation Identification task.

Setting / Model	Link Detection			Relation Classif.		
	Yes	No	M-F ₁	Sup	Att	M-F ₁
<i>Stand-Alone</i>						
Gemma (Zero-Shot)	.28	.96	.62	.30	.00	.15
Gemma (Detailed)	.22	.98	.60	.23	.00	.11
Gemini (Detailed)	.46	.99	.73	.48	.00	.23
<i>Sequential Evaluation</i>						
Gemma (Zero-Shot)	.25	.95	.60	.25	.00	.12
Gemma (Detailed)	.27	.95	.61	.26	.00	.13
Gemini (Detailed)	.48	.97	.73	.48	.09	.29
<i>End-to-End Pipeline</i>						
Gemma (Zero-Shot)	.22	.97	.59	.22	.00	.11
Gemma (Detailed)	.20	.97	.58	.20	.00	.10
Gemini (Detailed)	.45	.98	.71	.46	.06	.26

4.2 Pipeline and Evaluation

We evaluate our agents under three distinct configurations:

Stand-alone Agents. Each agent is evaluated separately against the ground truth, allowing us to measure the performance of individual agents without the confounding effects of upstream errors.

Sequential Evaluation. Agents are executed sequentially and each one is evaluated only on instances where the preceding agents performed correctly. For example, the PTC agent is evaluated on sentences that are both annotated as premises in the gold standard and classified as such by the ACC agent. The purpose of this setting is to evaluate each agent individually in a pipeline setting, without penalizing it for mistakes made by its predecessors. However, it inherently reduces the evaluation sample size for deeper steps in the pipeline, based on upstream accuracy.

End-to-end Pipeline. The evaluation is performed on the downstream predictions of the pipeline. For example, if a factual premise

is erroneously classified as a conclusion by the ACC agent, the PTC agent will not consider it a premise, and therefore it will not be labeled as either legal or factual. This will result in a false negative for the factual class and a true negative for the legal class. In contrast, if a conclusion is erroneously classified as a premise by the ACC agent, the PTC agent will be asked to classify it. If it classifies it as legal, it will be considered a false positive for the legal class and a true negative for the factual class. The purpose of this setting is to evaluate the performance of the pipeline as a whole on the final result, without distinguishing the contribution of single agents.

4.3 Models

We experiment with Gemma 4 26B, an open-weight Mixture-of-Experts model, and Gemini 3.1 Pro, a larger commercial model. They both have built-in reasoning capabilities and configurable thinking effort, which we leave at the default configuration.

We evaluate Gemma using two approaches. In the Zero-Shot approach, the prompts contain the definitions of the classes and rules. In the Detailed approach, the prompts also contain selected text segments extracted from the non-test part of the dataset to highlight distinctive linguistic patterns and textual cues. This approach is similar to Few-Shot prompting, since we include examples from the dataset to enrich the prompts, but we do not include the entire text of such examples. Given our limited resources, to optimize API costs, we evaluate Gemini exclusively under the latter configuration.

5 Results

We evaluate each experimental setting for each task by measuring the F_1 score on each class and the Macro-averaged F_1 (M-F₁). The experimental results for the first four tasks across the three evaluation configurations and models are reported in Table 2. Table 3 reports the results of the relation identification task, broken down into its two sub-components: Link Detection and Relation Classification.

In the Stand-Alone setting, Gemini (Detailed) consistently outperforms both Gemma configurations across the first three tasks.

PTC is the task where the best result is obtained, while the ASC task is the most difficult for all models. Underrepresented classes like Classification and Principle yield poor results, with Gemma (Zero-Shot) getting the best result. Interestingly, while the use of Detailed prompting with Gemma lowers the score in the ACD and ASC tasks, it significantly improves ACC.

Link Detection is a significantly challenging task. In the Stand-Alone setting Gemini outperforms Gemma with a score of .46 and .28 on the positive class, respectively. Interestingly, Gemma (Zero-Shot) outperforms its Detailed counterpart, suggesting that the provided examples may have introduced noise or led the model to overfitting to superficial patterns. As concerns Relation Classification, all the models completely fail to identify the Attack relation.

Moving from the Stand-Alone to the Sequential Evaluation setting, Gemma's performance degrades in the first four tasks. Instead, Gemini's results remain consistent. The End-to-End Pipeline configuration reveals significant performance degradation in all models, likely due to cascading errors in all five tasks. However, the intensity of this effect varies by model scale, with Gemini exhibiting significantly higher robustness. Indeed, Gemini retains moderate performance, while Gemma's scores are significantly lower. Counterintuitively, performance degradation within the End-to-End pipeline does not increase across sequential tasks for the most robust model, Gemini. The performance gap narrows from 15 points in ACC to 10 points in PTC and 7 points in ASC. A possible explanation could be the multi-label nature of the latter tasks: components missed or misclassified in early stages are excluded from downstream classifications. These unassigned labels, when evaluated as true negatives, artificially increase the metrics, despite upstream errors.

6 Conclusion

In this paper, we evaluated the performance of state-of-the-art LLMs on several complex tasks of Legal Argument Mining with a multi-agent LLM pipeline. Our findings demonstrate that both Gemini and Gemma are significantly affected by error propagation in an end-to-end pipeline, with Gemma showing higher vulnerability. Future work could explore more advanced MAS paradigms, such as multi-agent debate and iterative self-reflection loops, allowing agents to provide feedback and structurally correct upstream classification errors before finalizing the argument graph.

Ethics and Privacy Statement

All the data used in this work are publicly available. We envision our methods as tools that support and empower legal practitioners in their activities, and not as tools that perform autonomously without any human supervision.

References

- [1] Francesco Alfieri, Giulia Grundler, Francesca Galloni, Ruta Liepina, Francesca Lagioia, Andrea Galassi, and Paolo Torroni. 2025. Dynamic Demonstrations Selection for Few-Shot Legal Argument Mining. In *AMELR@ICAIL (CEUR Workshop Proceedings)*. CEUR-WS.org, 15–29.
- [2] Christina Berghegger, César Philippe, Karla Salas-Jimenez, Jean-Guy Mailly, Leila Moudjari, and Laurent Perrussel. 2025. Discovering the Potential of LLMs in Annotating Legal Texts for Argument Mining. In *Arg&App 2025*. 33–44.
- [3] Deniz Gorur, Antonio Rago, and Francesca Toni. 2025. Can Large Language Models perform Relation-based Argument Mining?. In *COLING*. Association for Computational Linguistics, 8518–8534.
- [4] Morgan A. Gray, Li Zhang, and Kevin D. Ashley. 2025. Generating Case-Based Legal Arguments with LLMs. In *CSLAW*. ACM, 160–168.
- [5] Giulia Grundler, Andrea Galassi, Piera Santin, Alessia Fidelangeli, Federico Galli, Elena Palmieri, Francesca Lagioia, Giovanni Sartor, and Paolo Torroni. 2024. AMELIA - Argument Mining Evaluation on Legal documents in ItAlia: A CALAMITA Challenge. In *CLiC-it (CEUR Workshop Proceedings, Vol. 3878)*. CEUR-WS.org.
- [6] Giulia Grundler, Piera Santin, Andrea Galassi, Federico Galli, Francesco Godano, Francesca Lagioia, Elena Palmieri, Federico Ruggeri, Giovanni Sartor, and Paolo Torroni. 2022. Detecting Arguments in CJEU Decisions on Fiscal State Aid. In *ArgMining@COLING*. International Conference on Computational Linguistics, 143–157.
- [7] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large Language Model Based Multi-agents: A Survey of Progress and Challenges. In *IJCAI*. ijcai.org, 8048–8057.
- [8] Cong Jiang and Xiaolei Yang. 2023. Legal Syllogism Prompting: Teaching Large Language Models for Legal Judgment Prediction. In *ICAIL*. ACM, 417–421. doi:10.1145/3594536.3595170
- [9] Huihao Jing, Wenbin Hu, Hongyu Luo, Jianhui Yang, Wei Fan, Haoran Li, and Yangqiu Song. 2025. MASLegalBench: Benchmarking Multi-Agent Systems in Deductive Legal Reasoning. *CoRR* abs/2509.24922 (2025).
- [10] Ruta Liepina, Francesca Galloni, Francesca Lagioia, Marco Lippi, Mariaceleste Musico, Burcu Sayin, Andrea Passerini, and Giovanni Sartor. 2025. Legal Argument Mining: Recent Trends and Open Challenges. In *AMELR@ICAIL (CEUR Workshop Proceedings, Vol. 4089)*. CEUR-WS.org, 1–14.
- [11] Raquel Mochales Palau and Marie-Francine Moens. 2009. Argumentation mining: the detection, classification and structure of arguments in text. In *ICAIL*. ACM, 98–107.
- [12] Prakash Poudyal, Jaromír Savelka, Aagje Ieven, Marie-Francine Moens, Teresa Gonçalves, and Paulo Quaresma. 2020. ECHR: Legal Corpus for Argument Mining. In *ArgMining*. Association for Computational Linguistics, 67–75.
- [13] Petros Raptopoulos, Giorgos Filandrianos, Maria Lymperaiou, and Giorgos Stamou. 2025. PAKTON: A Multi-Agent Framework for Question Answering in Long Legal Agreements. In *EMNLP*. Association for Computational Linguistics, Suzhou, China, 7948–7984. doi:10.18653/v1/2025.emnlp-main.403
- [14] Piera Santin, Giulia Grundler, Andrea Galassi, Federico Galli, Francesca Lagioia, Elena Palmieri, Federico Ruggeri, Giovanni Sartor, and Paolo Torroni. 2023. Argumentation Structure Prediction in CJEU Decisions on Fiscal State Aid. In *ICAIL*. ACM, 247–256.
- [15] Cor Steging, Ludi van Leeuwen, and Tadeusz Zbiegień. 2026. Investigating Multi-Agent Deliberation in Law. In *AIDA2J Workshop*. <https://arxiv.org/abs/2606.30906>
- [16] William Watson, Nicole Cho, Nishan Srishankar, Zhen Zeng, Lucas Cecchi, Daniel Scott, Suchetha Siddagangappa, Rachneet Kaur, Tucker Balch, and Manuela Veloso. 2025. LAW: Legal Agentic Workflows for Custody and Fund Services Contracts. In *COLING (Industry)*. Association for Computational Linguistics, 583–594.
- [17] Hiroaki Yamada, Simone Teufel, and Takenobu Tokunaga. 2019. Building a corpus of legal argumentation in Japanese judgement documents: towards structure-based summarisation. *Artif. Intell. Law* 27, 2 (2019), 141–170.
- [18] Abdullah Al Zubaer, Michael Granitzer, and Jelena Mitrovic. 2023. Performance analysis of large language models in the domain of legal argument mining. *Frontiers Artif. Intell.* 6 (2023).