

ArguGenCourt: Courtroom Argument Generation using Judicial Multi-agent Large Language Models

Subinay Adhikary

subinayadhikary612@gmail.com

Indian Institute of Science Education and Research Kolkata

Gourish Majamdur

Indian Institute of Technology Guwahati

Saurab Mishra

Indian Institute of Science Education and Research Thiruvananthapuram

Debasis Ganguly

University of Glasgow

Dwaipayan Roy

Indian Institute of Science Education and Research Kolkata

Souvik Kumar Guha

West Bengal National University of Juridical Sciences

Kripabandhu Ghosh

Indian Institute of Science Education and Research Kolkata

Research Article

Keywords: Legal Argumentation, Courtroom Simulation, Multi-Agent Large Language Models, Argument Quality Evaluation

Posted Date: October 26th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10788123/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

ARGUGENCOURT: Courtroom Argument Generation using Judicial Multi-agent Large Language Models

Abstract

In the legal domain, *judgment prediction* is an important and challenging problem. In real-life proceedings, judicial decisions depend on the precise presentation of complex case information and, above all, on the argumentative interplay among petitioners, respondents, and judges. However, most existing research work on this judgment prediction primarily focuses on case facts, rather than on argumentative legal reasoning. Furthermore, many such approaches rely on single-agent Large Language Models (LLMs), which are not analogous to the real-life courtroom. To address this limitation, simulating a courtroom conversation becomes essential. Some prior studies explored multiple LLMs (multi-agent) based courtroom simulations, with limited investigation on the quality of argumentation. Motivated by this crucial gap, in this work, we propose simulating courtroom arguments employing a *multi-agent* conversational system, termed as ARGUGENCOURT, specifically tailored to crime-related cases. We consider solely legal cases from the *U.S. Supreme Court*, evaluating the ability of LLMs in generating courtroom arguments using different in-context learning approaches, including zero-shot, and the variant of few-shot techniques, within a multi-agent setup. Empirically, we find that the quality of arguments improves significantly when using *local precedent retrieval* based approach employing GPT-4o models. This strategy yields 12% improvement over the zero-shot (baseline) setup. Subsequently, to measure the quality of LLM-based arguments, we propose a new algorithm – coined as **ArguEval**. Eventually, to showcase the efficacy of generated arguments, we utilize them in a downstream **judgment prediction** task, which results in a 14% performance gain over the zero-shot baseline.

Keywords: Legal Argumentation, Courtroom Simulation, Multi-Agent Large Language Models, Argument Quality Evaluation

1 Introduction

Large Language Models (LLMs) have been widely used in the legal domain for different downstream tasks, such as judgment prediction [1, 2], statute prediction [3, 4], rhetorical rule identification [5–7], and legal attribute extraction [8].

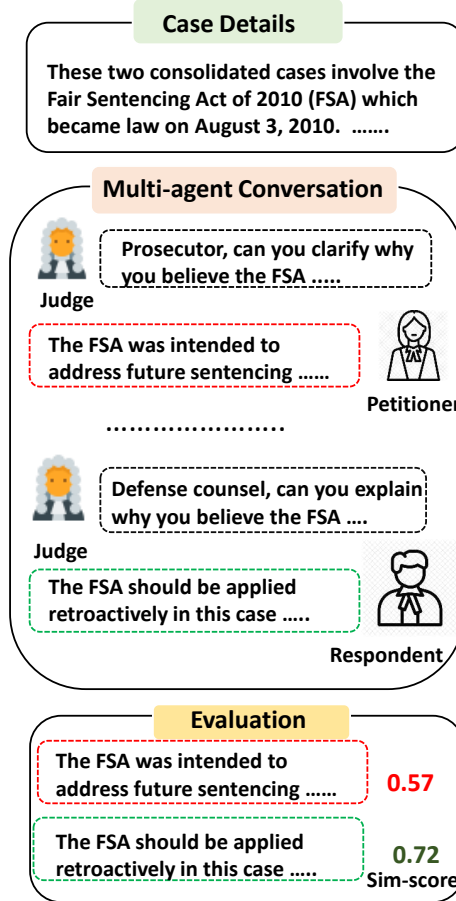


Fig. 1 This figure shows LLM-based multi-agent conversation, *judge* agent communicates with *petitioner* and *respondent*, based on provided details of the case. Later, we evaluate the arguments generated by the *petitioner* and *respondent* agents using existing metrics (e.g., Bert-score) in comparison with real-life courtroom arguments.

Subsequently, different in-context learning strategies such as zero-shot, few-shot, and chain-of-thought (CoT), are tailored for judgment prediction along with reasoning solely based on facts of the case [4, 8]. However, this setup does not mirror real-world courtroom proceedings, where judicial decisions are shaped not only by case facts and precedents but also by the dynamic interactions among the *judge*, *petitioner*, and *respondent*.

Recent studies [9–11] have leveraged multiple LLMs in a multi-agent framework to simulate a courtroom environment, primarily using Chinese legal datasets. However, these prior works have concentrated on downstream tasks such as judgment prediction and argument generation. However, they have *not* attempted to validate: *Are LLMs capable of faithfully simulating real-world courtroom interactions by accurately representing the roles of legal experts?* We argue that analyzing the quality of generated

conversation is important, as non-relevant conversation, even with high prediction accuracy of a downstream task, questions the efficacy of the conversational system. In this study, we deploy multiple LLMs, or *agents*, which collaborate to simulate a courtroom environment (shown in **Figure 1**). More specifically, we focus on generating contextually appropriate arguments that closely resemble real-life courtroom conversation, within an LLM-based multi-agent setup, coined as ARGUGENCOURT. In **Figure 2**, we showcase that multiple agents (e.g., *judge*, *petitioner* and *respondent*) are seen to be communicating with each other, and to investigate arguments more closely, we take a subtle approach by focusing on analyzing the performance of LLMs in generating arguments from several aspects, as detailed in Section 4.3. We then leverage these generated arguments for *judgement prediction* to measure the effectiveness. We believe that this experimental framework offers a more realistic simulation of courtroom dynamics and can help practicing advocates prepare for unforeseen counterarguments in ongoing or upcoming litigation. Additionally, it holds significant pedagogical value for training law students in realistic courtroom scenarios.

Contributions:

In this paper, we make the following contributions:

- **Exploration of Argument Generation Approaches:** We investigate three different methods for generating arguments in a multi-agent framework, as outlined in Section 4.3.
- **Utterance-Based Few-Shot Selection:** Our methods rely on selecting utterances as few-shot examples to generate arguments for both petitioners and respondents.
- **Novel Multi-Agent Setup:** To the best of our knowledge, this is the first study that systematically examines the effectiveness of multi-agent LLMs within a collaborative courtroom environment.
- **Evaluation Framework — ArguEval:** We propose a new evaluation algorithm, ArguEval, to assess the relevance and quality of generated arguments effectively, as detailed in Section 5.

All the codes and data developed for this research are publicly available for reproducibility and research purposes¹.

2 Related Work

This section reviews existing literature on LLM-based human simulation and LLM-based agent systems applied to legal reasoning.

2.1 LLM-based Human Simulation

Recently, researchers have begun leveraging LLM-based multi-agent systems to tackle increasingly complex tasks. For example, [12] introduces an LLM-driven collaborative framework for meta-programming, while [13] designs multiple LLM agents to facilitate both high-level communication and low-level path planning among robots. Given

¹https://anonymous.4open.science/r/Multi_agent-1EBD/

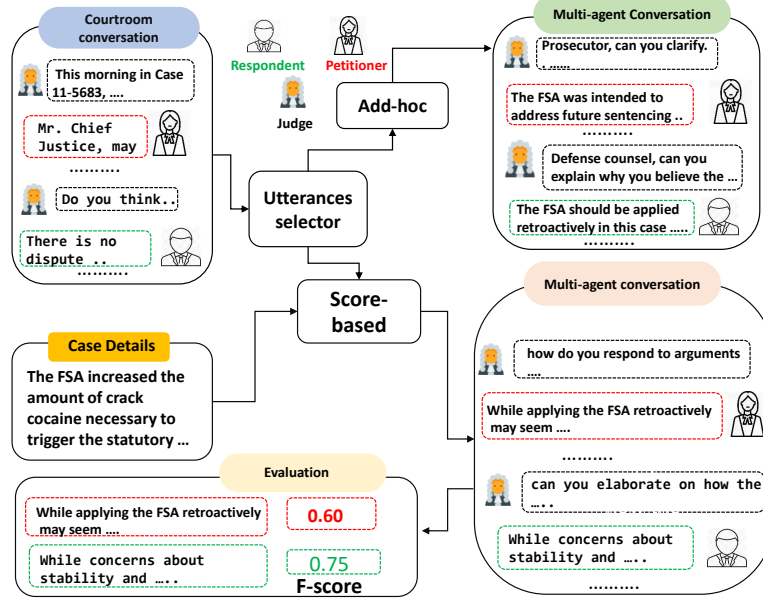


Fig. 2 We show that, for selecting the utterances from the same document, we explore two approaches: random selection (ad-hoc) and score-based selection. In the score-based selection, we provide the case-specific closest utterances as examples, namely called *Local Precedent Retrieval* (LPR). Subsequently, ARGUGENCOURT produces two sets of arguments: one between judge and petitioner and the other between judge and respondent agent. Finally, we evaluate the generated arguments for both the petitioner and respondent.

that LLMs inherently capture human-like behaviors through large-scale training data, an expanding body of research now employs them for human scenario simulation, exploring their potential as LLM-Empowered Agents capable of reasoning, decision-making, and interaction across diverse domains — including social and psychological studies [14–19], software engineering [12, 20], chemistry and medicine [21, 22], and gaming [23]. Furthermore, recent works [24–27] propose innovative collaboration strategies to strengthen coordination and performance within multi-agent frameworks. Collectively, these advancements offers for applying multi-agent LLM systems in the legal domain.

2.2 LLM-based Agent Systems in Law

In the domain of court simulation, Chen et al. [10] introduced a simplified trial debate simulation framework, AgentCourt, where lawyer agents evolve through adversarial debates. Similarly, He et al. [9] proposed AgentsCourt, a judicial decision-making agent that integrates trial debate simulation with legal knowledge augmentation, highlighting the significance of modeling the trial process for the Legal Judgment Prediction (LJP) task. In contrast, Zhang et al. [28] overlooked the systematic design and evaluation of simulation processes, thereby limiting their applicability in legal education and decision support. Hamilton [11] trained nine distinct models based on

the authored opinions of individual Supreme Court judges, providing valuable insights into judicial behavior; however, it only predicts the judges’ votes without capturing the underlying complexity of the decision-making process. To accounting this issue, we first evaluate the quality of LLM-generated arguments to *address the question of their faithfulness*. Subsequently, we leverage these generated arguments for judgment prediction.

3 ARGUCOURT: A Framework for Courtroom Argument Generation

In this section, we describe the principal components of our proposed setup: ARGUCOURT comprising *courtroom agents*, *memory module*, *communication strategies*, and provide a detailed overview of each component’s role and functionality.

ARGUCOURT consists of *three* courtroom agents, namely **judge**, **petitioner** and **respondent**, each modeling a key participant in courtroom proceedings:

- **Judge Agent:** This agent assumes the role of the judicial authority, overseeing the proceedings, posing clarifying questions, evaluating the presented arguments, and potentially rendering decisions based on the deliberation.
- **Petitioner Agent:** This agent represents the party initiating the case. The petitioner presents arguments, submits supporting evidence and reasoning, and responds to challenges raised by the respondent.
- **Respondent Agent:** This agent represents the defending party. The respondent formulates counter-arguments, challenges the petitioner’s claims, and defends against the allegations presented.

Each of the three agents is powered by an independent Large Language Model (LLM), enabling the following capabilities:

- **Independent Language Processing:** Each agent possesses its own language understanding and generation capabilities.
- **Role-Specific Reasoning:** Agents formulate responses according to their designated courtroom roles and objectives.
- **Natural Language Argumentation:** The architecture supports sophisticated, multi-turn argumentative interactions.
- **Role-Aware Prompting:** Each LLM is prompted and configured in alignment with the agent’s judicial function.

To maintain contextual awareness across interactions, every agent is equipped with a `ConversationBufferMemory`² module from the LangChain framework, which provides the following functionalities:

- **Conversation History Storage:** Maintains a complete record of all interactions occurring during the courtroom session.
- **Context Preservation:** Enables agents to retain and utilize prior conversational context for coherent responses.

²<https://docs.langchain.com/oss/python/langgraph/memory>

- **Information Retrieval:** Allows agents to reference previously stated arguments, facts, or claims.
- **Conversational Continuity:** Ensures logical flow by reducing unnecessary repetition and internal contradictions.

As a result, conversational continuity is maintained, reducing unnecessary repetition and preventing contradictions with earlier positions. During each conversational turn, an agent receives structured input from another agent, which is processed by its LLM in conjunction with the stored conversational memory. Inter-agent communication is coordinated through the `ConversationChain` component of LangChain, which orchestrates the dialogue as follows:

- **Sequential Processing:** During each conversational turn, an agent receives input from another agent in the form of a structured prompt.
- **Response Generation:** The receiving agent processes the prompt using its LLM while leveraging the memory module for contextual grounding.
- **Chain Management:** `ConversationChain` manages information flow, ensuring proper turn-taking and message passing among agents.
- **Prompt Engineering:** Messages are automatically transformed into role-appropriate prompts for the receiving agent.

Based on previous discussion and facts of the case, *judge* and *respondent* agents engage in an argument. In both cases, *petitioner* and *respondent* receive the *judge*'s questions and respond. The judge then reads their response and formulates follow-up questions accordingly. During the k rounds of conversations, where $k \in [1, 3, 5]$, agents remember the conversations they had in previous rounds and answer/generate the next questions, considering it, as detailed in *LangGraph*³. This prevents repetition of the same line of questioning in subsequent rounds and helps maintain a natural flow to the conversation. Eventually, ARGUGENCOURT generates two versions of argumentation: one between the *judge* and *petitioner* agents, and the other between *judge* and *respondent* agents. Both the input received and the response generated are stored in the memory. Later, we stored all the generated arguments in a JSON file. We examine the effectiveness of a *multi-agent* setup compared to a *single-agent* approach using in-context learning-based methods (see Section 4).

4 Experimental Setup

This section introduces the experimental dataset and details two strategies for utterance selection in the context of in-context learning.

4.1 Dataset

In this paper, we use publicly available U.S. Supreme Court documents⁴. The dataset covers cases from 2000 to 2019 and includes a total of 1,428 legal documents. In this study, we have focused on three categories of legal cases - *criminal*, *civil*, and

³<https://langchain-ai.github.io/langgraph/concepts/memory/>

⁴<https://convokit.cornell.edu/documentation/supreme.html>

Prompt Template in ArguGenCourt

You are an expert [ROLE]. You have completed your master’s in law and have been practicing in American court for over 10 years [ROLE_DESCRIPTION].

[INSTRUCTION]

Please keep your answer short and to the point. Answer in less than 2 sentences.

Example interaction between two legal roles:

examples: [EXAMPLE]

-----end of example-----

Case details: [DETAILS]

The Judge will now ask you questions. Here is the question asked by the judge:

[JUDGE_QUESTION]

Here is the next question asked by the judge: [JUDGE_QUESTION]. Answer within 2 sentences.

constitutional; a sample of 450 documents in total, with 150 documents from each categories are selected. From this collection, we select 100 documents as the training set S , and use the remaining documents as the test set T . Each document includes the following components: (i) the case facts, (ii) courtroom conversations, referred to as *utterances*, between the judge, the petitioner, and the respondent, (iii) the final judgment, and (iv) the reasoning underlying the judgment.

The average fact description contains approximately 250 tokens, providing a concise summary of the case background and a solid foundation for legal argumentation. As these facts are sourced from U.S. Supreme Court cases and have been used in real courtroom proceedings, we consider them sufficiently reliable and informative for our study. Accordingly, we incorporate them into the in-context learning setup of our empirical evaluation.

Each petitioner’s argument contains an average of 80 tokens, while judge and respondent arguments average 40 and 75 tokens, respectively. On average, each case comprises around 300 utterances, considering both. As shown in Table 4, the 300 selected documents contain a total of 97,237 utterances, which are used both as few-shot examples.⁵

We employ GPT-4o⁶ and Llama-3.1-70b⁷ for argument generation. We ultimately report the results from GPT-4o in Table 6, as it achieves the best performance.

⁵A subset of the dataset is publicly available at <https://anonymous.4open.science/r/Multi-agent-404E/>. We plan to release the entire set upon acceptance.

⁶<https://platform.openai.com/docs/models/gpt-4o>

⁷<https://huggingface.co/meta-llama/Llama-3.1-70B>

Prompt for Courtroom Argumentation between Judge and Petitioner

You are an expert [Advocate]. You have completed your master’s in law and have been practicing in American court for over 10 years. In this case you are to act as the prosecutor advocate. [Your job is to make sure that your client the victim accused is convicted and is sent to jail.]

[As the system goes, the judge is going to ask you questions about the case. You are to answer questions trying to prove that the accused is indeed guilty.]

Please keep your answer short and to the point. Answer in less than 2 sentences.

Here is an example of some questions and answers between judge and prosecutor: examples:

Judge: We’ll hear argument next in Case 18-877, Allen versus Cooper. Mr. Shaffer.

Petitioner: Mr. Chief Justice, and may it please the Court:

.....

—————end of example—————

Here are the case details:

“In 1996, a private researcher hired petitioner Frederick Allen and his company, Nautilus Productions, ”

The Judge will now ask you questions. Here is the question asked by the judge:

“... would be highly persuasive if we didn’t have the patent decision ...”

Here is the next question asked by the judge: {Judge_Question} answer within 2 sentences.

4.2 Baseline

In this setup, we provide facts of the case to begin the dialogue between *judge* and *petitioner* agents. Later, the petitioner’s arguments are utilized along with facts in the conversation between *judge* and *respondent* agents. Hence, this approach solely depends on the facts of the case. We call this method **Fact-based Conversation** or **FC** in short.

4.3 Similarity-Based Document Matching for Few-Shot

According to the legal professionals we consulted, they prepare current case dialogue based on similar prior case proceedings, where courtroom arguments are given for the respective proceedings. Following the same concept, we provide examples of utterances along with facts of the case, which aids *petitioner* and *respondent* agents to improve accuracy and minimize the risk of deviating from the relevant topics.

Given a test set $S = \{D_s^1, D_s^2, \dots, D_s^m\}$ and a training set $T = \{D_t^1, D_t^2, \dots, D_t^n\}$, our objective is to identify a relevant training document $D_t^* \in T$ that can serve as a source of representative and contextually similar examples for the test documents. These examples are subsequently incorporated into a few-shot prompting framework to guide the behavior of LLM-based agents.

Prompt for Courtroom Argumentation between Judge and Respondent

You are an expert [Advocate]. You have completed your masters in law and have been practicing in American court for over 10 years. In this case you are to act as the prosecutor advocate. [Your job is to make sure that your client the victim accused is convicted and is sent to jail.]

[As the system goes, the judge is going to ask you questions about the case. You are to answer questions trying to prove that the accused is indeed guilty.]

Please keep your answer short and to the point. Answer in less than 2 sentences.

Here is an example of some questions and answers between judge and prosecutor: examples:

Judge: Ms. Millett, can I ask you this one question? Would there ever be a case where it was not harmless error when the judge makes the finding?

Respondent: Mr. Chief Justice, and may it please the Court– The Washington Supreme Court correctly held that as a matter of Federal and State law..

.....

—————end of example—————

Here are the case details:

“In 1996, a private researcher hired petitioner Frederick Allen and his company, Nautilus Productions,”

The Judge will now ask you questions. Here is the question asked by the judge:

“... would be highly persuasive if we didn’t have the patent decision ...”

Here is the next question asked by the judge: {Judge-Question} answer within 2 sentences.

Table 4 Statistical Overview of U.S. Supreme Court Oral Arguments

Dataset		Legal Case Type		
Type	Statistics	Crime	Civil	Constitutional
Train	# Documents	100	100	100
Train	# Utterances	33342	31432	32463
Test	# Documents	50	50	50
Test	# Utterances	16342	15652	17234

For each selected document D_t^* , we extract a set of utterances that closely align with the factual content of the test document(s). These utterances are embedded into the prompt as illustrative examples, enabling the model to better capture domain-specific reasoning patterns and discourse structure. Table 3 presents the prompt templates used for the three different agents considered in our framework, while Table 3 illustrates an example interaction between the *judge* and the *petitioner*, demonstrating

how the retrieved utterances are integrated within the prompt. And, Table 3 illustrates an example interaction between the *judge* and the *respondent*, demonstrating how the retrieved utterances are integrated within the prompt

To identify the most suitable training document D_t^* , we propose two complementary document selection strategies, described below.

Precedent Similarity Maximization (PSM)

The *Precedent Similarity Maximization* strategy selects a single training document that is globally relevant to the entire test set for each legal category. Specifically, it identifies the training document that maximizes the cumulative fact-level semantic similarity with all test documents within each legal type. Formally, this objective is defined as:

$$D_t^* = \arg \max_{D_t^j \in T} \sum_{i=1}^m \text{sim}(D_t^j, D_s^i), \quad (1)$$

where $\text{sim}(D_t^j, D_s^i)$ denotes the fact-level semantic similarity between the training document D_t^j and the test document D_s^i . This similarity is computed using contextual embeddings generated by LegalBERT for both documents, allowing the model to capture nuanced legal semantics beyond surface-level lexical overlap.

Once the optimal document D_t^* is identified, a subset of utterances is randomly sampled from it to construct few-shot examples. This global selection strategy ensures consistency across prompts and is particularly suitable when the test set exhibits substantial thematic overlap. The overall workflow of this approach is illustrated in Figure 2.

Local Precedent Retrieval (LPR)

In contrast to PSM, the Local Precedent Retrieval strategy adopts a more fine-grained, instance-specific approach by selecting a distinct training document for each individual test document. For a given test document $D_s^i \in S$, we retrieve the training document $D_t^* \in T$ that exhibits the highest fact-level semantic similarity. This selection process is formalized as:

$$D_t^* = \arg \max_{D_t^j \in T} \text{sim}(D_s^i, D_t^j), \quad (2)$$

where $\text{sim}(D_s^i, D_t^j)$ represents the fact-level semantic similarity between the test document D_s^i and the training document D_t^j , again computed using LegalBERT embeddings.

After identifying the most relevant training document for a given test instance, we select utterances from D_t^* that are most closely aligned with the factual content of D_s^i . This is achieved by ranking utterances in D_t^* based on their similarity scores with the facts in D_s^i and selecting the top-scoring utterances. As shown in Figure 2, this localized retrieval strategy allows the few-shot prompts to be tailored to individual test documents, thereby improving contextual relevance and potentially enhancing downstream agent performance.

Another way to generate relevant arguments is by replacing example utterances when the generated judge’s question is not relevant to the case from the same case document. However, its implementation relies on repeatedly measuring similarity between

<p>Petitioner 11: The judges of the Seventh Circuit are unanimous in their belief that this case ...</p> <p>Petitioner 12: The primary holding was based on the fact that there was a specific provision for no retroactivity...</p>	<p>Petitioner 21: The Fair Sentencing Act was intended to address ...</p> <p>Petitioner 22: Prior case law, such as <i>United States v. Johnson</i>, has established that statutes are generally not applied retroactively ...</p>
<p>Evaluation Score:</p> $\sigma(\mathcal{R}_p, \mathcal{L}_p) = \frac{1}{2}(\max\{\text{sim}(P21, P11), \text{sim}(P21, P12)\} + \max\{\text{sim}(P22, P11), \text{sim}(P22, P12)\}) \quad (3)$ $\sigma(\mathcal{R}_p, \mathcal{L}_p) = \frac{1}{2}(0.56 + 0.65) = 0.6050$	

Table 5 Real-life Courtroom Arguments ($\mathcal{R}$) vs LLM-based Multi-agent Generation ($\mathcal{L}$). An example in this table is used to illustrate the evaluation method we propose.

generated questions and case-specific content, and dynamically retrieving suitable utterances from the same document after each conversational turn. However, selecting examples from the same document may introduce data leakage and limit applicability when such contextual resources are unavailable. These constraints pose significant challenges for adopting this *guided few-shot* strategy in realistic legal argument generation systems.

5 Evaluation

In this section, we introduce a novel strategy for evaluating the quality of generated arguments, termed as **ArguEval**.

5.1 Traditional Similarity Metrics

To compute the similarity function $\theta(L_p^i, R_p^j)$ between a simulated argument L_p^i and a real argument R_p^j , we investigate both lexical and semantic similarity metrics.

Lexical Similarity Metrics

We initially employ lexical similarity metrics, including ROUGE-L and METEOR, which measure word-level overlap between arguments. However, these metrics yield relatively low similarity scores (typically less than 0.2), suggesting substantial surface-level mismatches between the simulated and real arguments. This observation indicates that while the generated arguments may capture the semantic intent, they do not necessarily reproduce the exact wording used in real courtroom conversations.

Table 6 Comparison of in-context learning strategies across single-agent and multi-agent configurations for Petitioner and Respondent using GPT-4o on Criminal, Murder, and Constitutional cases. BS, LS, and BL denote BERTScore, Legal-sim, and BLEURT, respectively. **Bold** entries indicate the best performance.

Model	Party	Agent	Method	Criminal			Murder			Constitutional		
				BS	LS	BL	BS	LS	BL	BS	LS	BL
GPT-4o	Petitioner	Single	FC	0.4823	0.8025	0.4126	0.4921	0.8012	0.4014	0.4883	0.7888	0.4121
			PSM	0.4856	0.8176	0.4224	0.5069	0.8234	0.4232	0.5023	0.8256	0.4423
			LPR	0.5060	0.8225	0.4256	0.5213	0.8179	0.4278	0.5155	0.8423	0.4533
		Multi	FC	0.5123	0.8288	0.4381	0.5211	0.8267	0.4223	0.5234	0.8257	0.4335
			PSM	0.5267	0.8456	0.4472	0.5335	0.8456	0.4468	0.5323	0.8456	0.4675
			LPR	0.5682	0.8921	0.4782	0.5723	0.8934	0.4813	0.5674	0.8910	0.4823
	Respondent	Single	FC	0.4940	0.8123	0.4278	0.4824	0.8021	0.4113	0.4923	0.8110	0.4274
			PSM	0.5025	0.8236	0.4356	0.4861	0.8021	0.4127	0.4961	0.8176	0.4287
			LPR	0.5235	0.8400	0.4388	0.4954	0.8424	0.5258	0.5209	0.8243	0.4514
		Multi	FC	0.5320	0.8512	0.4479	0.5314	0.8123	0.4612	0.5327	0.8334	0.4456
			PSM	0.5455	0.8653	0.4576	0.5521	0.8623	0.4538	0.5421	0.8522	0.4619
			LPR	0.5755	0.8924	0.4845	0.5816	0.8895	0.4823	0.5775	0.8916	0.4896

Semantic Similarity Metrics

To address the limitations of lexical metrics, we adopt semantic similarity measures that capture meaning beyond surface-level word matching. Specifically, we evaluate three semantic similarity metrics:

- **BERTScore** [29]: Computes similarity using contextualized embeddings from BERT, matching tokens based on their contextual representations.
- **BLEURT** [30]: A learned metric trained on human judgments that captures nuanced semantic similarity.
- **Legal-sim**: A domain-specific similarity metric computed using LEGAL-BERT [31] embeddings, which are pre-trained on large-scale European and U.S. legal corpora. For arguments L_p^i and R_p^j , we first obtain their LEGAL-BERT embeddings $\mathbf{e}_{L_p^i}$ and $\mathbf{e}_{R_p^j}$, then compute cosine similarity:

$$\text{Legal-sim}(L_p^i, R_p^j) = \frac{\mathbf{e}_{L_p^i} \cdot \mathbf{e}_{R_p^j}}{\|\mathbf{e}_{L_p^i}\| \|\mathbf{e}_{R_p^j}\|} \quad (4)$$

As reported in Table 6, the semantic similarity scores are substantially higher than the lexical scores, indicating that the simulated arguments are *semantically aligned with the ground-truth despite exhibiting lower verbatim overlap* (as demonstrated in Table 4.3). This finding validates that our multi-agent system successfully captures the argumentative intent and legal reasoning, even when the exact phrasing differs from real conversations.

5.2 ArguEval

To rigorously assess the quality of arguments generated by LLM-based multi-agent systems, we introduce a novel evaluation framework, termed *ArguEval*. To the best of

our knowledge, no existing evaluation tool is capable of reliably measuring the semantic fidelity and contextual relevance of simulated courtroom dialogues with respect to real-world courtroom conversations. *ArguEval* addresses this critical gap by systematically comparing generated arguments against authentic legal discourse, enabling a fine-grained assessment of argumentative coherence, relevance, and alignment with real-life courtroom reasoning. By grounding the evaluation in real courtroom interactions, our framework provides a principled and domain-aware mechanism for evaluating the realism and effectiveness of multi-agent conversational systems in legal settings.

In this section, we first formalize the problem definition, followed by a detailed description of *ArguEval*. We then present the key components of the proposed evaluation framework and explain how it is used to assess the quality and relevance of arguments generated by LLM-based multi-agent courtroom simulations.

Problem Formulation

Consider a test document D_s^i for which we have access to a real-life courtroom conversation between a *judge* and a *petitioner*. We represent this ground-truth conversation as an ordered sequence:

$$\mathcal{R} = \{R_j^1, R_p^1, R_j^2, R_p^2, \dots, R_j^k, R_p^k, \dots, R_j^m\} \quad (5)$$

where R_j^k denotes the k -th utterance by the judge and R_p^k denotes the k -th utterance by the petitioner in the real conversation. Using our LLM-based multi-agent framework, we simulate a conversation for the same document D_s^i , which generates the following sequence:

$$\mathcal{L} = \{L_j^1, L_p^1, L_j^2, L_p^2, \dots, L_j^t, L_p^t, \dots, L_j^n\} \quad (6)$$

where L_j^t and L_p^t represent the t -th utterances by the simulated judge and petitioner, respectively.

Argument-Level Evaluation Strategy

Since each argument in a courtroom dialogue conveys distinct and context-specific information, we adopt an argument-level evaluation approach rather than treating the entire conversation as a monolithic unit. This granular evaluation strategy allows us to assess the relevance and quality of individual arguments, as illustrated in Table 4.3.

Let $\mathcal{R}_p = \{R_p^1, R_p^2, \dots, R_p^m\}$ denote the set of all petitioner utterances from the real conversation, and let $\mathcal{L}_p = \{L_p^1, L_p^2, \dots, L_p^n\}$ denote the set of all petitioner utterances from the simulated conversation. Our evaluation proceeds as follows:

1. For each simulated argument $L_p^i \in \mathcal{L}_p$, we compute its similarity with all real arguments $R_p^j \in \mathcal{R}_p$ using a similarity function $\theta(\cdot, \cdot)$.
2. We identify the maximum similarity score for each simulated argument:

$$\max_{R_p^j \in \mathcal{R}_p} \theta(L_p^i, R_p^j) \quad (7)$$

This maximum score represents the best match between the simulated argument L_p^i and any argument in the ground-truth set.

3. We repeat this process for all simulated arguments in $\mathcal{L}_p$ and compute the average of these maximum similarity scores.

Formally, we define the evaluation score as:

$$\sigma(\mathcal{R}_p, \mathcal{L}_p) = \frac{1}{n} \sum_{i=1}^n \max_{R_p^j \in \mathcal{R}_p} \theta(L_p^i, R_p^j) \quad (8)$$

where $\sigma(\mathcal{R}_p, \mathcal{L}_p)$ represents the final evaluation score that quantifies how well the simulated arguments align with the ground-truth arguments. A higher score indicates better alignment between the simulated and real conversations.

In this work, we apply the *ArguEval* framework to evaluate both the *petitioner's* and *respondent's* arguments generated by our system. For each role (petitioner and respondent), we compare the simulated arguments against their corresponding ground-truth arguments using Equation 8. Let $\mathcal{R}_r = \{R_r^1, R_r^2, \dots, R_r^m\}$ denote the ground-truth respondent arguments and $\mathcal{L}_r = \{L_r^1, L_r^2, \dots, L_r^n\}$ denote the simulated respondent arguments. This comprehensive evaluation across both roles provides a holistic assessment of our multi-agent system's capability to generate legally sound and contextually relevant arguments.

5.3 Expert Evaluation

While automatic metrics (Section 5.2) provide a scalable means of assessing surface-level similarity between generated and reference arguments, they are inherently limited in their ability to capture legal soundness, argumentative coherence, and doctrinal correctness. Courtroom argumentation is a highly specialized form of discourse, governed by procedural conventions, jurisdiction-specific precedent, and rhetorical strategies that purely lexical or embedding-based metrics (e.g., BLEU, ROUGE, or BERTScore) cannot adequately capture. To address this limitation, we conducted a structured human evaluation study with domain experts to assess the qualitative fidelity of LLM-generated courtroom arguments against their corresponding original (ground-truth) arguments.

For each case instance, experts were presented with a triplet consisting of: (i) the case context and preceding judicial statement, (ii) the original human-authored argument, and (iii) the corresponding LLM-generated argument, with the presentation order randomized to prevent positional bias. Experts were asked to independently rate the degree of alignment between the generated and original arguments along legal substance, argumentative structure, and contextual relevance, without being informed which text was machine-generated.

The alignment between LLM-generated arguments and original courtroom arguments was scored on a discrete 5-point scale ranging from 0 to 4, as detailed below:

- **4 – Full alignment:** The generated argument closely mirrors the original in legal reasoning, factual grounding, and rhetorical intent, with no substantive deviation.

- **3 – Mostly aligned:** The generated argument captures the core legal position and reasoning of the original, with minor omissions or stylistic differences that do not alter the substantive claim.
- **2 – Partially aligned:** The generated argument reflects some elements of the original’s reasoning or claims but diverges in key legal points, structure, or emphasis.
- **1 – Minimally aligned:** The generated argument shares only superficial thematic overlap with the original, with limited correspondence in legal substance.
- **0 – Irrelevant or contradictory:** The generated argument is unrelated to the original or directly contradicts the legal position, facts, or intent expressed in the original argument.

Given the two-party adversarial structure of our simulated courtroom debate, we conducted evaluation separately for the Judge–Petitioner exchange and the Judge–Respondent exchange, as these differ in argumentative posture (affirmative claim-making versus defensive rebuttal) and may exhibit different failure modes.

To assess the reliability and consistency of expert judgments, multiple experts independently rated each case instance, and inter-annotator agreement was computed using weighted Cohen’s kappa. Given the ordinal structure of the 0–4 scale, we report agreement using a weighted agreement coefficient that penalizes larger disagreements (e.g., a 0 vs. 4 rating) more heavily than adjacent disagreements (e.g., a 2 vs. 3 rating). Cases with substantial disagreement among experts (i.e., a score range exceeding two points) were flagged for discussion and adjudicated through a consensus session to arrive at a final agreed-upon score, following standard practice in NLP human evaluation studies.

6 Results and Analysis

The results in Table 6 demonstrate the effectiveness of multi-agent strategies and fine-tuning across Criminal, Civil, and Constitutional domains for both Petitioner and Respondent. Across all metrics, *single-agent* models outperform *multi-agent* ones; for instance, in the Petitioner Criminal domain, LPR achieves a BERTScore of 0.5682 with multi-agent fine-tuning, compared to 0.5060 for single-agent pre-trained, representing an improvement of approximately 12.4%. Among methods, Local Precedent Retrieval (LPR) consistently achieves the highest scores, followed by PSM and FC, highlighting the importance of leveraging local precedents. Multi-agent setups consistently enhance performance over single-agent configurations; for Petitioner Criminal LPR, Legal-sim improves from 0.8225 to 0.8921 (+8.5%), and BLEURT from 0.4256 to 0.4782 (+12.4%), indicating that collaborative reasoning among agents effectively integrates factual and precedent information. Similar trends are observed for Respondent tasks, where multi-agent LPR achieves 0.5755 (BERTScore) in Criminal, 0.5816 in Civil, and 0.5775 in Constitutional domains, consistently exceeding single-agent scores by 5–10%. Domain-wise, Criminal tasks see the largest gains from multi-agent LPR, followed by Constitutional, with Civil tasks showing moderate but consistent improvements, particularly in BLEURT, reflecting better semantic alignment. Overall, these results confirm that multi-agent collaboration and effective precedent retrieval collectively yield substantial improvements—up to 12%—across all legal similarity

Table 7 Presents the F1-scores for both Class 0 and Class 1, as well as the accuracy of the judgment prediction task using *GPT-4* models. The best-performing values, highlighted in **bold**, indicate the effectiveness of the proposed (*LPR*) approach.

Method	Class 0	Class 1	Accuracy
0-shot	0.45	0.50	0.48
2-shot	0.48	0.52	0.50
CoT	0.50	0.56	0.53
LPR	0.52	0.58	0.55

metrics, demonstrating the effectiveness of the proposed strategies for legal reasoning and argument alignment.

Downstream Task: Judgment Prediction

To comprehensively evaluate the effectiveness and practical utility of the arguments generated by our ARGUGENCOURT framework, we conduct a series of judgment prediction tasks on 450 documents, where Class 0 and Class 1 are binary labels used to represent the outcome of a legal case. Specifically, *Class 0* indicates Judgment against the petitioner, and *Class 1* indicates Judgment in favor of the petitioner. The underlying hypothesis is that high-quality legal arguments should not only be semantically similar to ground-truth arguments (as measured by our *ArguEval* metric) but should also demonstrably improve downstream legal reasoning tasks. Specifically, we assess whether the generated arguments can enhance the performance of LLMs in predicting case outcomes—a critical task in legal AI that requires deep understanding of legal reasoning and argumentation.

We design our judgment prediction experiments using the same test document set $S = \{D_s^1, D_s^2, \dots, D_s^m\}$ employed in our argument generation evaluation. For each test document $D_s^i \in S$, the task is to predict the final judgment (e.g., accepted, rejected, or partially accepted) based on the available information about the case.

We establish multiple experimental configurations to systematically evaluate the contribution of generated arguments:

0-Shot Baseline (Fact-Only Prediction)

Our primary baseline represents a zero-shot learning scenario where the LLM receives only the case facts $F_{D_s^i}$ extracted from test document D_s^i without any additional context or examples. Formally, the prompt structure for this baseline is:

$$P_{0\text{-shot}}(D_s^i) = \{\text{Task Instruction}, F_{D_s^i}, \text{Output Format}\} \quad (9)$$

This configuration evaluates the LLM’s inherent capability to perform legal reasoning and judgment prediction based solely on its pre-trained knowledge and the facts of the case, without any exemplar demonstrations or generated arguments. The 0-shot baseline serves as our primary reference point for measuring improvement.

2-Shot Learning Setup

To investigate whether general few-shot learning can improve judgment prediction performance, we implement a 2-shot configuration. In this setup, the LLM is provided with two randomly selected training examples (D_t^i, Y_t^i) where $D_t^i \in T$ is a training document with its corresponding ground-truth judgment Y_t^i . The prompt structure becomes:

$$P_{2\text{-shot}}(D_s^i) = \{\text{Inst.}, (F_{D_t^1}, Y_t^1), (F_{D_t^2}, Y_t^2), F_{D_s^i}, \text{Out.}\} \quad (10)$$

The training examples are selected randomly without considering their relevance to the test case. This configuration helps us understand whether providing examples of the input-output mapping, independent of argument quality, contributes to improved predictions.

Chain-of-Thought (CoT) Reasoning

Inspired by recent advances in prompting strategies, we implement a Chain-of-Thought (CoT) approach where the LLM is explicitly instructed to generate intermediate reasoning steps before producing the final judgment. The prompt structure for CoT is:

$$P_{\text{CoT}}(D_s^i) = \{\text{Inst.}, \text{Trig.}, F_{D_s^i}, \text{Path}, \text{Out.}\}$$

where the CoT trigger is an instructional prompt (e.g., “*Let’s think step by step about this case and provide your reasoning before the final judgment.*”), encouraging the LLM to explicitly articulate its reasoning. This structured reasoning typically involves analyzing key facts and their legal implications, identifying relevant legal principles or precedents, weighing competing arguments, and producing a justified final judgment.

Argument-Guided Prediction with Local Precedent Retrieval

Finally, we leverage the arguments generated by our ARGUGENCOURT framework to guide the judgment prediction process. Specifically, we utilize the Local Precedent Retrieval (LPR) method, which identifies the most similar training document for each test case and generates contextually relevant arguments through our multi-agent system. We evaluate LPR-based prediction approach.

LPR (LPR): In this configuration, we use arguments generated by our multi-agent system where the underlying LLM (e.g., GPT-4, Claude) operates in its pre-trained state without any domain-specific fine-tuning. For test document D_s^i , we first identify the most relevant training document D_t^* using Equation 2, then generate arguments $\mathcal{A}_{D_s^i} = \{A_1, A_2, \dots, A_k\}$ through simulated courtroom dialogue. The judgment prediction prompt becomes:

$$P_{\text{LPR}}(D_s^i) = \{\text{Inst.}, F_{D_t^*}, \mathcal{A}_{D_s^i}, \text{Out.}\} \quad (11)$$

The key distinction between these argument-guided approaches and the baseline methods is that the LLM receives *explicitly generated arguments* that simulate the deliberative process of a real courtroom. These arguments provide structured reasoning, highlight relevant legal principles, and present multiple perspectives on the case, thereby offering richer context for judgment prediction.

Table 7 presents a comprehensive comparison of all experimental configurations across multiple evaluation metrics, including class-wise precision, recall, F1-scores, and overall accuracy. Our results reveal several important findings:

Superiority of Argument-Guided Prediction

The LPR method achieves the highest performance across all metrics, demonstrating that explicitly generated arguments provide substantial value for judgment prediction. Specifically, LPR outperforms:

- The 0-shot baseline by a significant margin, validating that generated arguments effectively augment the LLM’s reasoning capabilities.
- The 2-shot configuration, indicating that quality of examples (generated arguments) matters more than mere presence of examples.
- The CoT approach, suggesting that our structured multi-agent argumentation provides more comprehensive reasoning than self-generated reasoning chains.

Class-Wise Performance Analysis

Examining the class-wise performance metrics in Table 7, we observe that LPR demonstrates consistent improvements across all judgment categories (e.g., accepted, rejected, partially accepted). This balanced performance suggests that the generated arguments capture nuanced legal reasoning applicable to diverse case outcomes, rather than being biased toward predicting specific judgment types. The improvement is particularly pronounced for minority classes, where the 0-shot and 2-shot baselines struggle due to class imbalance.

This extrinsic evaluation complements our intrinsic evaluation using *ArguEval* (Section 5), providing converging evidence that the generated arguments are of high quality and practical utility. The improvement in downstream task performance demonstrates that our generated arguments are not merely superficially similar to ground-truth arguments but genuinely encode useful legal reasoning.

7 Conclusion and Future Work

In this paper, we introduced ARGUGENCOURT, a multi-agent, LLM-based conversational framework designed to examine the effectiveness and realism of argument generation for both the *petitioner* and the *respondent* in a simulated courtroom setting. Unlike prior work that primarily emphasizes downstream outcomes, such as judgment prediction, our study focuses on the quality, contextual relevance, and fidelity of the generated courtroom discourse itself. We investigated

multiple strategies for selecting documents and utterances that guide argument generation, highlighting the importance of case-aligned and role-specific contextual grounding. To closely approximate real-world courtroom dynamics, we assigned distinct legal roles to specialized LLM agents and fine-tuned them to simulate interactions among the **judge**, **petitioner**, and **respondent**. This role-aware, collaborative setup enables structured argumentation that more closely mirrors authentic legal proceedings. Furthermore, we proposed a novel evaluation framework, **ArguEval**, to systematically assess the relevance, coherence, and legal soundness of generated

arguments. By leveraging judgment prediction as a downstream validation task, we demonstrated that higher-quality conversational arguments contribute meaningfully to improved predictive performance, reinforcing the necessity of evaluating conversational realism alongside task accuracy.

Overall, our findings suggest that faithful simulation of courtroom interactions requires moving beyond isolated reasoning over case facts toward modeling multi-party legal discourse. We believe that ARGUGENCOURT lays an important foundation for future research in realistic legal AI systems, with practical implications for assisting legal practitioners in preparing for counterarguments in ongoing or upcoming litigation. Additionally, this framework holds significant promise as a pedagogical tool for training law students through exposure to realistic, interactive courtroom scenarios. In future work, we aim to extend this framework to broader legal domains, incorporate richer legal knowledge bases and precedents, and explore human-in-the-loop evaluations to further bridge the gap between simulated and real-world courtroom practices.

References

- [1] Feng, Y., Li, C., Ng, V.: Legal judgment prediction: A survey of the state of the art. In: IJCAI, pp. 5461–5469 (2022)
- [2] Nigam, S.K., Sharma, A., Khanna, D., Shallum, N., Ghosh, K., Bhattacharya, A.: Legal judgment reimaged: Predex and the rise of intelligent ai interpretation in indian courts. arXiv preprint arXiv:2406.04136 (2024)
- [3] Paul, S., Bhatt, R., Goyal, P., Ghosh, S.: Legal statute identification: A case study using state-of-the-art datasets and methods. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR ’24, pp. 2231–2240. Association for Computing Machinery, New York, NY, USA (2024). <https://doi.org/10.1145/3626772.3657879> . <https://doi.org/10.1145/3626772.3657879>
- [4] Vats, S., Zope, A., De, S., Sharma, A., Bhattacharya, U., Nigam, S., Guha, S., Rudra, K., Ghosh, K.: Llms—the good, the bad or the indispensable?: A use case on legal statute prediction and legal judgment prediction on indian court cases. In: Findings of the Association for Computational Linguistics: EMNLP 2023, pp. 12451–12474 (2023)
- [5] Bhattacharya, P., Paul, S., Ghosh, K., Ghosh, S., Wyner, A.: Identification of rhetorical roles of sentences in indian legal judgments. In: Araszkievicz, M., Rodríguez-Doncel, V. (eds.) Legal Knowledge and Information Systems - JURIX 2019: The Thirty-second Annual Conference, Madrid, Spain, December 11–13, 2019. Frontiers in Artificial Intelligence and Applications, vol. 322, pp. 3–12. IOS Press, ??? (2019). <https://doi.org/10.3233/FAIA190301> . <https://doi.org/10.3233/FAIA190301>

- [6] Bambroo, P., Adhikary, S., Bhattacharya, P., Chakraborty, A., Ghosh, S., Ghosh, K.: Marro: multi-headed attention for rhetorical role labeling in legal documents. *Artificial Intelligence and Law*, 1–30 (2025)
- [7] Malik, V., Sanjay, R., Guha, S.K., Hazarika, A., Nigam, S., Bhattacharya, A., Modi, A.: Semantic segmentation of legal documents via rhetorical roles. *arXiv preprint arXiv:2112.01836* (2021)
- [8] Adhikary, S., Sen, P., Roy, D., Ghosh, K.: A case study for automated attribute extraction from legal documents using large language models. *Artificial Intelligence and Law*, 1–22 (2024)
- [9] He, Z., Cao, P., Wang, C., Jin, Z., Chen, Y., Xu, J., Li, H., Liu, K., Zhao, J.: Agentscourt: Building judicial decision-making agents with court debate simulation and legal knowledge augmentation. In: *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 9399–9416 (2024)
- [10] Chen, G., Fan, L., Gong, Z., Xie, N., Li, Z., Liu, Z., Li, C., Qu, Q., Ni, S., Yang, M.: Agentcourt: Simulating court with adversarial evolvable lawyer agents. *arXiv preprint arXiv:2408.08089* (2024)
- [11] Hamilton, S.: Blind judgement: Agent-based supreme court modelling with gpt. *arXiv preprint arXiv:2301.05327* (2023)
- [12] Hong, S., Zheng, X., Chen, J., Cheng, Y., Wang, J., Zhang, C., Wang, Z., Yau, S.K.S., Lin, Z., Zhou, L., et al.: Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352* (2023)
- [13] Mandi, Z., Jain, S., Song, S.: Roco: Dialectic multi-robot collaboration with large language models. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 286–299 (2024). IEEE
- [14] Aher, G.V., Arriaga, R.I., Kalai, A.T.: Using large language models to simulate multiple humans and replicate human subject studies. In: *International Conference on Machine Learning*, pp. 337–371 (2023). PMLR
- [15] Park, J.S., O’Brien, J., Cai, C.J., Morris, M.R., Liang, P., Bernstein, M.S.: Generative agents: Interactive simulacra of human behavior. In: *Proceedings of the 36th Annual Acm Symposium on User Interface Software and Technology*, pp. 1–22 (2023)
- [16] Li, G., Hammoud, H., Itani, H., Khizbullin, D., Ghanem, B.: Camel: Communicative agents for” mind” exploration of large language model society. *Advances in Neural Information Processing Systems* **36**, 51991–52008 (2023)
- [17] Li, N., Gao, C., Li, M., Li, Y., Liao, Q.: Econagent: large language model-empowered agents for simulating macroeconomic activities. *arXiv preprint*

arXiv:2310.10436 (2023)

- [18] Gao, C., Lan, X., Lu, Z., Mao, J., Piao, J., Wang, H., Jin, D., Li, Y.: S3: Social-network simulation system with large language model-empowered agents. arXiv preprint arXiv:2307.14984 (2023)
- [19] Zhang, J., Xu, X., Zhang, N., Liu, R., Hooi, B., Deng, S.: Exploring collaboration mechanisms for llm agents: A social psychology view. arXiv preprint arXiv:2310.02124 (2023)
- [20] Qian, C., Liu, W., Liu, H., Chen, N., Dang, Y., Li, J., Yang, C., Chen, W., Su, Y., Cong, X., et al.: Chatdev: Communicative agents for software development. arXiv preprint arXiv:2307.07924 (2023)
- [21] Li, J., Lai, Y., Li, W., Ren, J., Zhang, M., Kang, X., Wang, S., Li, P., Zhang, Y.-Q., Ma, W., et al.: Agent hospital: A simulacrum of hospital with evolvable medical agents. arXiv preprint arXiv:2405.02957 (2024)
- [22] M. Bran, A., Cox, S., Schilter, O., Baldassari, C., White, A.D., Schwaller, P.: Augmenting large language models with chemistry tools. *Nature Machine Intelligence* **6**(5), 525–535 (2024)
- [23] Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., Anandkumar, A.: Voyager: An open-ended embodied agent with large language models. arXiv preprint arXiv:2305.16291 (2023)
- [24] Cheng, Y., Liu, W., Wang, J., Leong, C.T., Ouyang, Y., Li, W., Wu, X., Zheng, Y.: Cooper: Coordinating specialized agents towards a complex dialogue goal. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 17853–17861 (2024)
- [25] Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., et al.: Autogen: Enabling next-gen llm applications via multi-agent conversations. In: *First Conference on Language Modeling* (2024)
- [26] Hu, Z., Chan, H.P., Li, J., Yin, Y.: Debate-to-write: A persona-driven multi-agent framework for diverse argument generation. In: *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 4689–4703 (2025)
- [27] Gorur, D., Rago, A., Toni, F.: Retrieval and argumentation enhanced multi-agent llms for judgmental forecasting. arXiv preprint arXiv:2510.24303 (2025)
- [28] Zhang, K., Li, J., Wu, Y., Li, H., Luo, C., Zou, S., Zhou, Y., Su, W., Ai, Q., Liu, Y.: Chinese court simulation with llm-based agent system. arXiv preprint arXiv:2508.17322 (2025)

- [29] Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675 (2019)
- [30] Sellam, T., Das, D., Parikh, A.P.: Bleurt: Learning robust metrics for text generation. arXiv preprint arXiv:2004.04696 (2020)
- [31] Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: Legal-bert: The muppets straight out of law school. arXiv preprint arXiv:2010.02559 (2020)