CASE: Conflict-Aware Synthesis of Evidence

Shreyans Babel
sbabel@umass.edu
University of Massachusetts
Amherst, MA, USA

Azeem Haider
mahaider@umass.edu
University of Massachusetts
Amherst, MA, USA

Ananya Srivastava
ananyasrivas@umass.edu
University of Massachusetts
Amherst, MA, USA

Bharat Govil
bgovil@umass.edu
University of Massachusetts
Amherst, MA, USA

Nithin Jeyanthinathan
njeyanthinat@umass.edu
University of Massachusetts
Amherst, MA, USA

Helly Dhamesha
hdhamesha@umass.edu
University of Massachusetts
Amherst, MA, USA

ACM Reference Format:
Shreyans Babel, Azeem Haider, Ananya Srivastava, Bharat Govil, Nithin Jeyanthinathan, and Helly Dhamesha. 2025. CASE: Conflict-Aware Synthesis of Evidence. In . ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Abstract

Conventional Retrieval-Augmented Generation (RAG) systems struggle with legal queries involving conflicting precedents and laws, often producing factually inaccurate answers. We investigate whether a multi-agent RAG pipeline that retrieves evidence for competing hypotheses can better ground answers in legal reasoning. We introduce *CASE: Conflict-Aware Synthesis of Evidence*, a novel multi-agent architecture that models the adversarial nature of legal reasoning by simulating courtroom logic through three stages: Hypothesis Generation, Advocate Retrieval, and Arbitrer Synthesis. A panel of legal personas evaluates conflicting evidence to select a majority consensus answer, reflecting diverse perspectives. Experiments on the Bar Exam QA dataset suggest that standard RAG systems often rely heavily on internal knowledge rather than retrieved context, highlighting the need for a conflict-aware, multi-agent approach like CASE to improve evidence-grounded reasoning in legal applications.

2 Problem statement

Conventional Retrieval-Augmented Generation (RAG) pipelines are optimized to find a single, authoritative answer. This approach is fundamentally misaligned with many real-world queries, which often involve unsettled law or conflicting precedents where no single correct answer exists. In these common legal scenarios, a standard RAG system is forced into one of two failure modes:

(1) **Misleading Oversimplification**: The system may retrieve evidence supporting only one side of a legal argument, presenting it as the definitive answer, and ignoring the broader conflict.
(2) **Factual Hallucination**: The system may attempt to resolve the ambiguity on its own, synthesizing a "resolution" that is factually incorrect, ungrounded, or simply misrepresents the legal query.

This leads us to our research question:

Can a multi-agent RAG pipeline that models legal ambiguity by retrieving evidence for competing hypotheses produce answers with demonstrably higher factual accuracy and grounded-ness compared to a standard, single-path RAG pipeline?

In our paper, we argue that to build a reliable legal AI framework, the system must embrace legal conflict rather than avoid it. We propose a novel multi-agentic architecture **CASE: Conflict Aware Synthesis of Evidence** that internalizes the adversarial process inherent in the legal field.

3 Related work

Multi-Agent Frameworks The **L-MARS** [8] paper utilizes a multi-agent workflow where a Query, Search, Judge and summary Agent collaborate to process a legal query. We propose adopting this multi-agentic paradigm, focusing on creating specialized agents that orchestrate complex reasoning and retrieval tasks. **HyPA-RAG** [5] uses a query complexity classifier to tune retrieval parameters (top-k and query rewrites). Their architecture combines dense, sparse, and knowledge graph retrieval methods to improve accuracy, demonstrating that a "one-size-fits-all" retrieval strategy is suboptimal for handling legal queries. LLMs are critically hampered by factual inaccuracy ("Hallucinations") and outdated knowledge. While RAG has emerged as one of the primary strategies to ground LLM outputs, recent research [6] demonstrates that standard RAG pipelines are insufficient for legal reasoning. **EMULATE** [3] also discusses multi-agent frameworks for determining veracity of true-false statements and provide credence to the concept that focused, single-task agentic frameworks can provide improvements over prior works. Our project builds on a confluence of these recent work that recognizes these limitations.

Despite claims of "hallucination-free" performance, the reliability of legal AI tools remains a major concern. In a first-of-its-kind empirical evaluation, Magesh et al. [6] assessed leading AI legal research tools from providers like LexisNexis and Thomson Reuters. They concluded these RAG-based systems hallucinate between 17% and 33% of the time. RAG, while an improvement over general-purpose chatbots, has not solved the hallucination problem in law [6].

Adversarial Agents The merit of multi-agent pipelines with agents working against each other's interest and its effect on answer accuracy is still being explored, with **AgentCourt**[2] providing evidence for agent improvement through adversarial evolution. **Mitigating Manipulation** [9] implements a similar reflective

CS 646 Information Retrieval,

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

multi-agent systems, where agents critique legal arguments additively. We extend the notion provided by both of these papers to multi-agent systems with conflicting agents.

Benchmarks A core obstacle in developing effective legal Retrieval-Augmented Generation (RAG) systems is the lack of benchmarks that capture the complexity of real-world legal research. Zheng et al.'s [10] Bar Exam QA and Housing Statute QA datasets show that legal retrieval is often a reasoning problem, requiring a connection between a factual query and a legal document with low lexical overlap. Simple lexical methods like BM25 struggle significantly, while performance improves with query expansion that encodes structured legal reasoning. Similarly, Hou et al.'s [4] CLERC, a colossal dataset for U.S. case law retrieval, demonstrates that even state-of-the-art models struggle with legal information retrieval, and Large Language Models (LLMs) are prone to hallucination when generating legal analysis.

4 Proposed Approach

CASE is a three-stage, multi-agent pipeline, divided into three distinct stages: Hypothesis Generation, Advocate Retrieval, and Arbiter Synthesis. Figure 1 visualizes our approach.

(1) **Hypothesis Generation (f_{hypo}):** An initial **Analyst Agent** (implemented by the `HypothesisGenerator` class) processes the user's query (q) alongside the multiple-choice options ($\{a, b, c, d\}$). The agent is engineered to execute one of two distinct strategies for formulating the initial hypothesis set (H):
   (a) **Direct Claim Adoption** (use generated hypotheses=False): The agent adopts a conservative strategy, treating the four provided MCQ options as the competing legal claims. The hypothesis set H is the content of the options.

$$H = \{h_1, \ldots, h_4\} = \{a, b, c, d\}$$

   (b) **Exploratory Hypothesis Generation** (use generated hypotheses=True): The agent employs an **LLM-driven exploratory approach**. It develops hypotheses to answer a downstream task that does not contain answers as options, based solely on the prompt and legal context. This shifts the burden to validating **root causes or principles** relevant to the question.

$$H = \{h_1, \ldots, h_n\} = f_{\text{LLM-gen}}(q)$$

   The resulting set H of competing claims then serves as the input for the Advocacy Stage.

(2) **Advocate Retrieval (f_{adv}):** For each hypothesis $h_i \in H$, we spawn an autonomous 'Advocate' agent with the explicit goal of constructing the strongest possible legal argument for h_i. It is a goal-oriented component equipped with a suite of tools, including vector search, keyword/sparse search (e.g., BM25), and LLM-based query formulation. The agent acts in a reasoning loop: it first analyzes h_i, then strategically selects and uses its tools to iteratively search the corpus D, reflect on the retrieved snippets, refine its search strategy, and finally compile the set E_i of the k most compelling evidence snippets.

$$E_i = \{e_{i1}, \ldots, e_{ik}\} = f_{\text{adv}}(h_i, D)$$

(3) **Arbitrer Synthesis (f_{synth}):**
To resolve the conflict generated by the advocates, and to minimize hallucinations, we implement a Persona based Jury Consensus Mechanism [7]. The final decision is reached through a structured evaluation carried out by a set of diverse legal agents.
We define a pool of $N = 5$ distinct legal personas, each prompted to evaluate evidence through a specific Jurisprudential lens:
- **The Strict Textualist**: This agent analyses provided evidence only and does not use any outside knowledge, rejecting any options not explicitly supported by the retrieved text and rigorously flagging any hallucinations.
- **The Legal Realist:** Focuses on the practical consequences. This agent evaluates whether the application of a rule leads to a sensible result, explicitly rejecting legal interpretations that would produce absurd results.
- **The Precedent Loyalist:** Prioritizes consistency. This agent compares the facts in the query strictly against the facts in the retrieved evidence and prevents false analogies where a rule is applied to a mismatched situation.
- **The Equity Advocate:** Views the law as a tool for fairness. In cases involving vulnerable parties, this agent interprets ambiguities to prevent unjust outcomes.
- **The Devil's Advocate:** Adopts a skeptical stance to find loopholes. This agent actively checks for exceptions, missing conditions, and weaknesses in the evidence presented by the Advocates.

At inference time, the system randomly samples a panel of $k = 3$ distinct personas from this pool. Each juror j independently reviews the user's query q and the complete set of hypothesis-evidence pairs. Using a Chain of Thought prompting strategy, each juror generates a reasoning trace explaining their decision based on their specific persona and casts a vote $v_j \in A, B, C, D$. The final synthesized answer a_{final} is determined by the majority consensus of this set.

$$a_{\text{final}} = \text{mode}\left(\{\, v_j \mid j \in \text{sampled\_personas} \,\}\right)$$

4.1 Baselines

To ground the evaluation of *CASE*, we implement and evaluate six distinct baseline models. These are all executed using a standardized experimental setup, which includes a unified client for generative models and a common answer-parsing module to ensure consistent scoring. The generative model used is meta-llama/llama-3.3-70b-instruct with a temperature of 0.3, except in the case of L-MARS.

Baseline 1: LLM-Only (Vanilla Prompting). This zero-shot baseline provides the question and options directly to the generator (`meta-llama/llama-3.3-70b-instruct`) with no retrieved context. This measures the model's out-of-the-box knowledge.

Baseline 2: BM25-RAG (Sparse RAG). This baseline represents a standard lexical RAG pipeline. We use `pyserini` to retrieve the

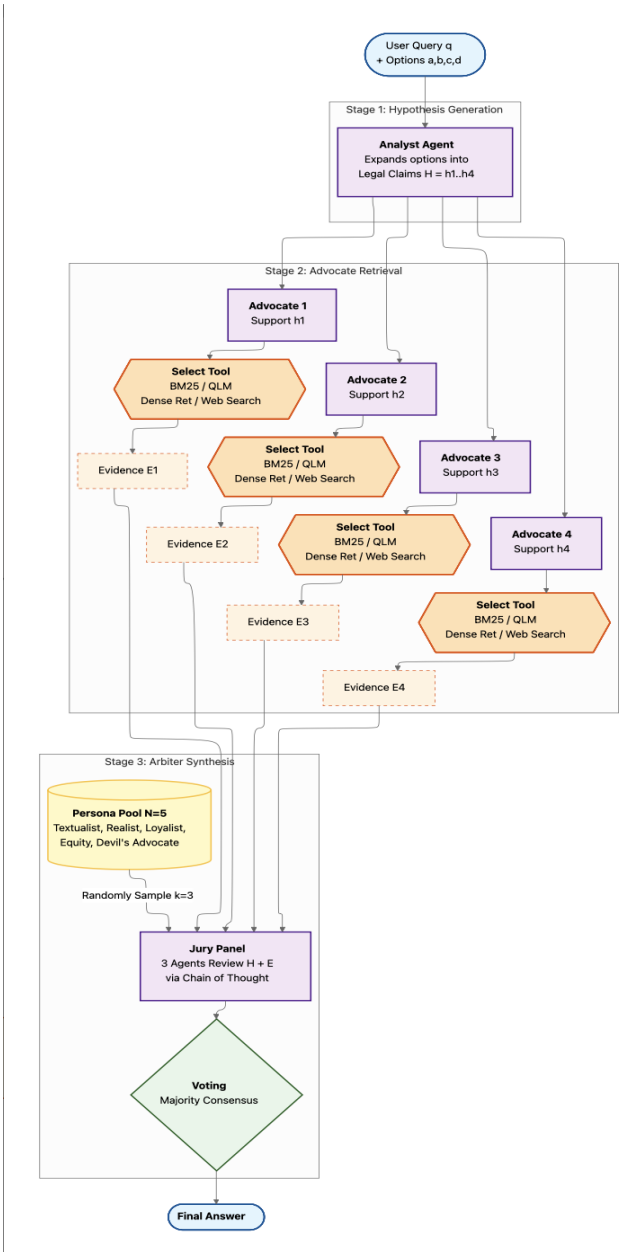

Figure 1: The CASE Architecture

top-k passages from a Lucene index of the passage corpus. These passages are concatenated and provided as context to the generator.

Baseline 3: QLM-RAG (Sparse RAG). This baseline uses a probabilistic retrieval model. We implemented a Query Likelihood Model (QLM) from scratch, ranking by $P(Q \mid D)$ with Dirichlet smoothing ($\mu = 2000$). The top-k passages are used as context for the generator.

Baseline 4: Standard Single-Path RAG (Dense RAG). This baseline represents a modern, semantic, single path, RAG pipeline. We use `BAAI/bge-small-en-v1.5` (384-dim) to build a `faiss.IndexFlatL2` index of the passage embeddings. At inference, the query is embedded, and the top-k nearest neighbors are retrieved as context for the generator.

Baseline 5: Self-RAG. This baseline evaluates the Self-RAG framework from [1] using a fine-tuned Llama2-13B model. The goal is to measure its accuracy when applied directly to our task.

Baseline 6: L-MARS (Multi-Agent RAG). This baseline serves as a state-of-the-art multi-agent comparator. We adapted L-MARS [8] and configured it to use `gpt-5-mini` as its generator.

5 Experiments

5.1 Datasets

For the evaluation of our Agentic RAG system, we will be using the two novel legal benchmarks from (Zheng et al., 2025) [10]. The **Bar Exam QA**[1] dataset consists of multistate bar exam (MBE) questions. Each MBE item presents a new legal situation, poses a question about the legal issue it raises, and provides four possible answers, with the goal of identifying the correct one. The retrieval passage pool contains 900K passages.

Our experiments focus specifically on the **Bar Exam QA validation split** ($n = 124$). The retrieval corpus for all our RAG baselines is the associated **Bar Exam QA passages** dataset (856,835 passages)

Table 1: Dataset Statistics (Q = Query, P = Passage) [10]

Dataset	Total Number		Avg. Length	
	Q	P	Q	P
Bar Exam QA	1,195	856,835	172	131

5.2 Evaluation

Our evaluation framework for our approach focuses on answer accuracy rather than retrieval metric. In a multi-agent setting like CASE, retrieval is not a single, well-defined step but an adaptive behavior that varies across agents and turns, making metrics such as Recall@k ill-defined. This reasoning aligns with findings reported in the L-MARS paper.

- **Answer Accuracy (primary):** For the multiple-choice Bar Exam QA, this is the percentage of items where the model's parsed single-letter choice (A, B, C, or D) matches the ground-truth gold label.

5.3 Experimental Setup

All experiments are run on the Bar Exam QA validation split ($n = 124$). The generator for Baselines 1–4 (LLM-Only, BM25, QLM, Dense RAG) is `meta-llama/llama-3.3-70b-instruct`, with a sampling temperature of 0.3. The generator for Baseline 5 (L-MARS) is `gpt-5-nano`. For all single-path RAG variants in our baselines, we report results for $k \in \{3, 5, 10\}$ passages. These passages are concatenated and prepended to a unified multiple-choice prompt template.

[1] https://huggingface.co/datasets/reglab/barexam_qa

Preprocessing Baselines: To enable retrieval for our RAG baselines, we constructed three distinct indices from the Bar Exam QA `passages` corpus, employing different preprocessing methodologies appropriate for each retrieval model:

- **BM25:** For sparse lexical retrieval, a standard Lucene index was built directly from the raw passage texts using the Pyserini toolkit.
- **QLM:** To support the Query Likelihood Model, the corpus was preprocessed by lowercasing and tokenizing via whitespace splitting. We then computed and stored global collection statistics, including term and document frequencies, necessary for the Dirichlet smoothing ($\mu = 2000$) calculation during retrieval.
- **Single-Path RAG:** To support our semantic RAG baseline, we generated vector representations for all 856,835 passages using the **`BAAI/bge-small-en-v1.5`** sentence transformer model. These 384-dimensional embeddings were indexed using `faiss.IndexFlatL2` for efficient nearest-neighbor search. A corresponding metadata file was created to map FAISS index positions back to the original passage IDs, facilitating retrieval evaluation.

For our experiments, we implemented the agentic system using the DSPy framework. We use LLaMA 3 70B as the base model, performing inference via OpenRouter and NVIDIA's API. The retrieval tools leverage prebuilt indices from the baseline experiments: BM25 (via Lucene), QLM (via Lucene with Dirichlet smoothening), a dense retriever using all-MiniLM-L6-v2 embeddings, and web search via Serper. During inference, we employ a jury-based decision mechanism: three jurors (k=3) are sampled randomly for each query, and the majority vote determines the final answer. This panel size was chosen to ensure clear majority decisions while maintaining robustness.

5.4 Results

Baseline Results: Our baseline tests yield the accuracy scores shown in Table 2 for $k \in \{3, 5, 10\}$, covering all RAG baselines. These initial results confirm that the baseline implementations are functioning as expected and provide a reliable foundation for subsequent evaluations.

The L-MARS model operates using Google Search as its retrieval mechanism. Consequently, RAG-based scores cannot be reported for this model, as it does not retrieve passages from the same local corpus used by our other retrieval-based baselines. Since both L-MARS and the LLM-Only baselines do not perform any retrieval from the evaluation corpus, they are reported separately from the RAG models. This separation ensures a clear distinction between methods that depend on explicit document retrieval and those that rely solely on the language model's internal reasoning or external search interfaces. Notably, we are also unable to report recall for the CASE multi-agent model framework proposed here. In this framework, retrieval is not a single, fixed operation but rather an outcome of dynamic, autonomous decisions during execution. Agents may choose whether to retrieve, which tool to use (e.g., BM25, QLM, dense retrievers, or web search), how many times to query, and how to reformulate queries across turns. This eliminates a stable reference point for a "retrieval step" and makes it unclear what should count as a retrieved document, whether it is any document accessed, those passed to the generator, or only those that influenced the final output.

Table 2: Generation accuracy on Bar Exam QA validation set ($n = 124$).

Baselines	Acc@3	Acc@5	Acc@10
BM25-RAG	65.32	58.06	66.12
QLM-RAG	67.74	62.90	65.32
Single Path RAG	62.10	58.06	62.10
Baselines	**Acc.**		
L-MARS	65.00		
LLM-Only	65.32		
Self-RAG	37.90		

For retrieval evaluation, recall scores were computed for $k \in 10, 100, 1000$ across BM25, QLM, and Single Path RAG. The corresponding Recall results are presented in Table 3. We observe that the Recall scores are relatively low across our retrieval-based experiments. This outcome can be attributed to the fact that each query in our dataset is associated with only a single golden passage. Given the large size of our retrieval corpus (approximately 900,000 passages) the likelihood of retrieving the correct passage within the top ranks is correspondingly small, leading to limited recall performance.

As the recall scores are low, we suspect that the language model may be leveraging its internal knowledge rather than relying primarily on the retrieved context. This hypothesis is further supported by the observation that the accuracy scores remain comparable to those achieved by models without retrieval augmentation. The model's performance does not appear to depend strongly on retrieved passages, suggesting that its internal pretraining data may already contain sufficient information to answer many of the queries correctly.

This interpretation is further reinforced by the performance of the fine-tuned Self-RAG LLaMA2:13B model, which exhibits lower accuracy. The reduced performance may be due to the smaller size and relative weakness of this model within the LLaMA family.

Table 3: Recall score on Bar Exam QA (validation, $n = 124$)

Model	Recall@10	Recall@100	Recall@1000
BM25-RAG	0.00	0.0323	0.1048
QLM-RAG	0.00	0.00	0.0108
Single Path Rag	0.00	0.00	0.0484

***CASE* Results:**

The results for the CASE multi-agent architecture (shown in Table 2) demonstrate that increasing the retriever agent's tooling does not always yield monotonic improvement. The baseline model using BM25 and RAG achieved 66.9% accuracy, which is relatively similar to our baseline RAG model results at $k = 3$. Adding the web search alone caused a performance drop to 65.3%, suggesting some

Table 4: Accuracy of CASE model across different retrieval variants ($n = 124$ and between high and low confidencee judgements).

Model	Acc.	High conf.	Low conf.
CASE (BM25, RAG)	66.9	76.00	53.06
CASE (BM25, RAG, Web)	65.3	73.24	54.72
CASE (BM25, RAG, Web, QLM)	**68.5**	**72.37**	**62.50**

level of noise injection. However, the final variant, which included QLM alongside the other three tools, achieved the highest accuracy at 68.5%, indicating that QLM effectively synthesizes the retrieved evidence to maximize performance.

Our definition of juror confidence is based on the degree of inter-agent agreement: high confidence corresponds to unanimous agreement among all three juror agents on a specific advocated conclusion, whereas low confidence is defined by a two-to-one majority agreement. We observed that confidence metrics remained relatively stable across model variants until QLM was introduced as an available tool. This addition exhibited a notable impact: it had negligible influence on the accuracy of high-confidence, unanimously agreed-upon decisions, but it significantly enhanced accuracy in the more uncertain, low-confidence scenarios.

5.5 Analysis

Score for Vanilla Prompting with the Gold Passage: We conducted an experiment using the LLM of our choice for this project (meta-llama/llama-3.3-70b-instruct), providing the model with prompts that included the gold passage associated with each question. The rationale was that if the model had access to the passage claimed by the dataset to contain the answer, its accuracy should exceed 90%. Surprisingly, the LLaMA model achieved only **72.58% accuracy (90/124)**, considerably lower than expected.

To explore this further, we selected a subset of five questions that LLaMA answered incorrectly despite having the gold passage in the prompt, and evaluated them using the ChatGPT and Gemini reasoning models. Gemini correctly answered **4/5**, while ChatGPT managed only **2/5**. A deeper examination of the questions the models missed revealed that in some cases, the gold passages were *vague*: although they were related to the question, they did not provide a *direct answer* without more nuanced understanding of the legal domain. This suggests that simply providing the gold passage is insufficient for high accuracy; domain expertise and careful reasoning remain crucial.

Adding different jury types and personas: Legal reasoning is adversarial by nature. Therefore, using a single neutral synthesizer to answer a legal query is often insufficient, as it tends to average out conflicting interpretations rather than resolving them. We chose to implement a "Persona-based Jury" to model this adversarial review process. By creating a diverse panel of five distinct legal personas ($N = 5$), we force the system to evaluate evidence from competing viewpoints. The **Strict Textualist** ensures grounding by rejecting unverified claims; the **Precedent Loyalist** prevents false analogies by verifying factual alignment; the **Legal Realist** filters out practically absurd outcomes; the **Equity Advocate** resolves ambiguity in favor of fairness; and the **Devil's Advocate** counteracts confirmation bias by actively seeking loopholes. This rigorous cross-examination forces the model to produce a legally defensible argument that holds up under different standards of review, rather than simply outputting the most likely average answer.

Future Work: Future research could explore several directions to improve reasoning over legal domain MCQs and question answering. First, evaluating how smaller LLMs fine-tuned specifically on legal domain data perform with vanilla prompting could reveal whether domain specialization improves inference from gold passages. Second, investigating techniques such as chain-of-thought prompting or rationale generation may enhance the models' ability to extract answers from vague or implicit gold passages.
Another aspect worth exploring is the addition of a feedback loop to the CASE pipeline. A query expansion and iterative retrieval mechanism could be implemented wherein the advocates refine their search queries using feedback from the jury, leading to multi round retrieval that better aligns with the differing jury personas.

References

[1] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *ICLR* (2024).

[2] Guhong Chen, Liyang Fan, Zihan Gong, Nan Xie, Zixuan Li, Ziqiang Liu, Chengming Li, Qiang Qu, Hamid Alinejad-Rokny, Shiwen Ni, and Min Yang. 2025. AgentCourt: Simulating Court with Adversarial Evolvable Lawyer Agents. In *Findings of the Association for Computational Linguistics: ACL 2025*. Association for Computational Linguistics, Vienna, Austria, 5850–5865. doi:10.18653/v1/2025.findings-acl.304

[3] Spencer Hong, Meng Luo, and Xinyi Wan. 2025. EMULATE: A Multi-Agent Framework for Determining the Veracity of Atomic Claims by Emulating Human Actions. In *Proceedings of the Eighth Fact Extraction and VERification Workshop (FEVER)*. Association for Computational Linguistics, Vienna, Austria, 179–183. doi:10.18653/v1/2025.fever-1.13

[4] Abe Bohan Hou, Orion Weller, Guanghui Qin, Eugene Yang, Dawn Lawrie, Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. 2025. CLERC: A Dataset for U. S. Legal Case Retrieval and Retrieval-Augmented Analysis Generation. In *Findings of the Association for Computational Linguistics: NAACL 2025*. Association for Computational Linguistics, Albuquerque, New Mexico, 7898–7913. doi:10.18653/v1/2025.findings-naacl.441

[5] Rishi Kalra, Zekun Wu, Ayesha Gulley, Airlie Hilliard, Xin Guan, Adriano Koshiyama, and Philip Colin Treleaven. 2024. HyPA-RAG: A Hybrid Parameter Adaptive Retrieval-Augmented Generation System for AI Legal and Policy Applications. In *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*. Association for Computational Linguistics, Miami, Florida, USA, 237–256. doi:10.18653/v1/2024.customnlp4u-1.18

[6] Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D Manning, and Daniel E Ho. 2025. Hallucination-Free? Assessing the Reliability of Leading AI Legal Research Tools. *Journal of Empirical Legal Studies* 22, 2 (2025), 216–242.

[7] Tsz Fung Pang, Maryam Berijanian, Thomas Orth, Breanna Shi, and Charlotte S. Alexander. 2025. PersonaMatrix: A Recipe for Persona-Aware Evaluation of Legal Summarization. arXiv:2509.16449 [cs.CL] https://arxiv.org/abs/2509.16449

[8] Ziqi Wang and Boqin Yuan. 2025. L-MARS: Legal Multi-Agent Workflow with Orchestrated Reasoning and Agentic Search. arXiv:2509.00761 [cs.AI] https://arxiv.org/abs/2509.00761

[9] Li Zhang and Kevin D. Ashley. 2025. Mitigating Manipulation and Enhancing Persuasion: A Reflective Multi-Agent Approach for Legal Argument Generation. arXiv:2506.02992 [cs.AI] https://arxiv.org/abs/2506.02992

[10] Lucia Zheng, Neel Guha, Javokhir Arifov, Sarah Zhang, Michal Skreta, Christopher D. Manning, Peter Henderson, and Daniel E. Ho. 2025. A Reasoning-Focused Legal Retrieval Benchmark. In *Proceedings of the Symposium on Computer Science and Law (CSLAW '25)*. ACM, 169–193. doi:10.1145/3709025.3712219

Contributions

All the authors worked on the research, writing and editing of the reports.

- Ananya: Arbitrer Synthesis, CASE Architecture diagram
- Helly: QLM baseline, CASE pipeline variation with QLM
- Azeem: BM25 and Self-RAG baseline, stage-1 hypothesis generator, and analyzed the performance of vanilla prompting with gold passages
- Bharat: CASE pipeline implementation, CASE metrics, confidence analysis
- Nithin: Advocate Tools/Structure,Indexing,API Integration,Baseline Analysis, Preprocessing
- Shreyans: Vanilla Prompting, RAG with a dense retriever, and LMARS; stage-2 Advocate Agent: baseline agentic code and implementing dense retriever, BM25 retriever, and web search