Self-RAG Framework for Hallucination Mitigation detected by INSIDE in Legal Analysis

Kavya Sridhar, Keisha Mehta, Kedar Desai, Yash Vishe, Guruprasad Parasnis, Marcus Chang
DSC 261 Responsible Data Science

1 ILLUSTRATIVE EXAMPLES OF HALLUCINATION DETECTION

To demonstrate the practical working of our integrated approach, this section presents two representative examples drawn from distinct areas of legal reasoning. Each case illustrates how the INSIDE framework identifies hallucinations through analysis of a model's internal state dynamics, and how Self-RAG subsequently performs retrieval-driven self-reflection to refine and verify the final output. Together, they exemplify how neural uncertainty detection and reflective reasoning collaboratively enhance factual alignment and interpretability in high-stakes legal contexts.

1.1 Example 1: Constitutional Interpretation: "Right to Privacy"

- **Prompt:** "Does the U.S. Constitution grant an explicit right to privacy?"
- **Hallucinated Response:** The baseline model incorrectly stated: "The Third Amendment explicitly guarantees the right to privacy."
  This constitutes a classic legal hallucination; the U.S. Constitution contains no explicit reference to privacy, though the right has been judicially inferred through interpretation.
- **Detection with INSIDE:** Using the INSIDE (Internal State Inspection for Detection) framework, multiple candidate responses were sampled. The EigenScore, quantifying the dispersion of hidden-state activations, showed high variance, indicating semantic inconsistency among generations. Feature clipping was then applied to mitigate overconfident activations that masked uncertainty. The elevated EigenScore and clipped activations jointly flagged the output as a likely hallucination, preventing it from being surfaced to the user.
- **Correction with Self-RAG:** Once flagged, the Self-RAG mechanism was triggered for reflective verification. The model emitted a Retrieve = Yes control token and accessed curated constitutional sources, including Griswold v. Connecticut (1965). Retrieved evidence was labeled ISREL = Relevant, and through iterative reasoning cycles, the model's ISSUP reflection transitioned from Partially supported to Fully supported.
  The revised response correctly stated: "The Constitution does not explicitly mention a right to privacy, but the Supreme Court recognized it as implied through the First, Third, Fourth, Fifth, and Ninth Amendments."

This case illustrates how INSIDE preemptively detects uncertainty, while Self-RAG performs structured self-verification to ground the output in judicial precedent. The combination ensures doctrinal precision and prevents misrepresentation of constitutional principles, essential for legal reasoning reliability.

1.2 Example 2: Contract Law, Enforceability of Promises

- **Prompt:** "Can a promise made without payment be legally binding?"
- **Hallucinated Response:** The initial model asserted: "Every promise is enforceable under contract law." This statement overlooks the doctrine of consideration, a fundamental principle determining the enforceability of contractual obligations.
- **Detection with INSIDE:** The INSIDE framework detected instability in the model's reasoning via elevated EigenScore and high internal-state entropy across samples. This pattern indicated logical inconsistency among candidate completions. Feature clipping was again used to reduce false certainty, leading the system to flag the response as unreliable before user exposure.
- **Correction with Self-RAG:** Self-RAG initiated a retrieval sequence querying verified legal databases and contract law treatises. Through reflective tokens, the model identified passages from Hamer v. Sidway (1891) and relevant statutory commentaries. During self-evaluation, the model adjusted its internal judgments: ISSUP = Partially supported → Fully supported.
  The final grounded output stated: "A promise is enforceable only when supported by consideration, something of legal value exchanged between parties, with limited exceptions such as promissory estoppel."

This example underscores Self-RAG's capacity to integrate doctrinal exceptions within broader legal frameworks. By iteratively retrieving, reflecting, and regenerating, the model transitions from overgeneralized reasoning to contextually complete legal analysis, enhancing both accuracy and interpretability.

- **Note:**
  - ISREL stands for "Is Relevant", indicating whether the retrieved passage is relevant to the query.
  - ISSUP stands for "Is Supported", reflecting whether the model's generated claim is fully grounded in the retrieved evidence.

2 METHODOLOGY

For this project, our methodology was structured into three key phases. We began by preparing the necessary legal data, then built a specialized retrieval pipeline. Following that, we trained the Self-RAG generator and critic models.

2.1 Data Preparation

First, we had to collect and process the data needed for both retrieval and training. This involved two main tasks.

- **Corpus Ingestion and Chunking:** Our legal corpus consisted of documents containing text, source, and title fields. To prepare these for the retrieval system, we first ran a statistical analysis to understand document lengths. We then processed all documents using our DocumentChunker, configuring it for a 256-character chunk size with a 30-character overlap. This process broke down the large documents into smaller, semantically coherent segments suitable for vector embedding.

- **Q&A Dataset Partitioning:** We also prepared a parallel Q&A dataset containing legal questions and their reference answers. To create distinct sets for training and evaluation, we partitioned this dataset using 'sklearn.model_selection.train_test_split', reserving 20% of the data for our final testing.

2.2 Retrieval Pipeline

With the data prepared, we built the retrieval pipeline, which serves as the foundation of our RAG system.

- **Embedding Model:** We selected the sentence-transformers/all-mpnet-base-v2 model as our bi-encoder. This model generates the 768-dimensional vector embeddings for both our document chunks and the incoming user queries.
- **Vector Indexing:** We stored the corpus embeddings in a FAISS (Facebook AI Similarity Search) index. We specifically chose an IndexFlatIP index, which uses Inner Product for similarity search and stores the full-precision vectors for maximum accuracy.
- **Retrieval Process:** At runtime, our system embeds an incoming query using the same all-mpnet-base-v2 model. It then searches the FAISS index to retrieve the top-k most relevant document chunks based on their inner product similarity score.
- **Persistence:** To make the system efficient and avoid re-indexing, we designed the pipeline to be persistent. The final FAISS index is saved as a .faiss file, and the document store (which maps IDs to text) is saved as a .pkl file. This allows us to load the complete, indexed retriever directly from disk.

2.3 Self-RAG Training

The core of our work was training the Self-RAG model. Unlike a standard RAG, our model learns to actively assess the quality of its retrieved information and its own generated answer.

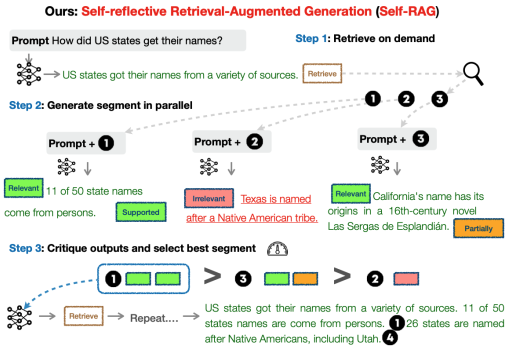
Ours: Self-reflective Retrieval-Augmented Generation (Self-RAG)
Prompt How did US states get their names?
Step 1: Retrieve on demand
US states got their names from a variety of sources.
Retrieve
Step 2: Generate segment in parallel
1
2
3
Prompt + 1
Prompt + 2
Prompt + 3
Relevant
11 of 50 state names come from persons.
Supported
Irrelevant
Texas is named after a Native American tribe.
Relevant
California's name has its origins in a 16th-century novel Las Sergas de Esplandián.
Partially
Step 3: Critique outputs and select best segment
1 > 3 > 2
Retrieve
Repeat....
US states got their names from a variety of sources. 11 of 50 states names are come from persons. 1 26 states are named after Native Americans, including Utah. 4

Figure 1: Self-RAG Architecture

- **Base Model:** We fine-tuned both the critic and generator components from the Qwen/Qwen2.5-1.5B-Instruct foundation model. Our first step was to expand the model's vocabulary with 18 new "reflection tokens" (e.g., [Relevant], [NoSupport]) that the model would learn to output.
- **Critic Model Training:** We trained the critic model first. We used a rule-based script (src.training.generate_labels) to pre-label our training Q&A dataset with the correct reflection tokens. We then fine-tuned the critic model on this labeled data using QLoRA (Quantized Low-Rank Adaptation). This process taught the model to predict the relevance of retrieved passages and the factual consistency of generated answers.
- **Generator Model Training:** Next, we trained the generator, also with QLoRA. For this stage, we augmented the training data with predictions from the critic we had just trained. This was the key step to teach the generator to produce both the final answer and the appropriate reflection tokens. Finally, we merged the generator and critic weights into a single, unified model capable of both generation and self-reflection.

3 EVALUATION

Finally, we needed to validate our system's performance. We used a two-pronged evaluation strategy to test the retrieval and generation components.

(1) **Retrieval Evaluation:** We benchmarked the retrieval pipeline on its own. Using a test set of queries with known ground-truth document IDs, we computed standard Information Retrieval (IR) metrics: Precision@k, Recall@k, Mean Reciprocal Rank (MRR), and Mean Average Precision (MAP).

(2) **Generation Evaluation:** We then evaluated the full end-to-end Self-RAG system using our held-out Q&A test set. Here, we focused on factual accuracy, hallucination reduction, and overall answer quality. Our key metrics included Hallucination Rate, FactScore, Completeness, and a Utility Score derived from the model's own reflection tokens.

This multi-stage process allowed us to robustly build and validate each component of our system for the complex legal domain.

4 RESULTS & ANALYSIS

After running our retrieval system against the LegalBench-RAG benchmark, our team found a very clear split in performance. The results present a "good news, bad news" story. The good news is that our system is highly effective at the document level. The bad news, and our critical challenge moving forward, is a significant failure at the snippet level.

- **Strong Precision and Recall:** Looking at the Metrics Curves, our Document Precision@1 is 55.93%. This is a strong start, as it means we're finding the correct document more than half the time on the very first try. Our Document Recall@k (top-right chart) shows steady growth, reaching 84.41% by k=64. This tells us that the relevant information is, in fact, present in the documents we're pulling the vast majority of the time. [Fig.2]
- **Snippet-Level Failure:** The snippet-level charts are almost completely flat and near zero. Our Snippet P@1 is 0.39% and our Snippet R@64 is only 0.27%. This is a major concern. It means we're successfully pulling the right contract, but the specific text chunk we retrieve is not the precise answer snippet the benchmark is looking for. Our team's analysis is that this is a chunking problem, not an embedding problem. Our current 256-character chunking strategy is clearly too broad and does not align with the

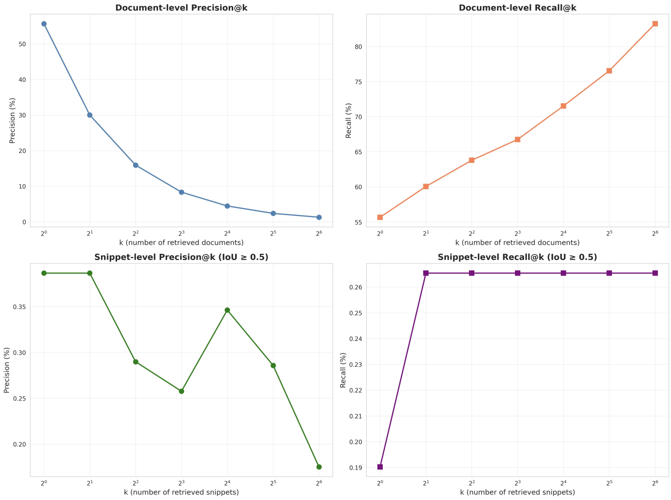

Figure 2: Document-level and Snippet-level Precision-Recall Curves

granular, sentence-level answers in the LegalBench-RAG ground truth. [Fig.2]

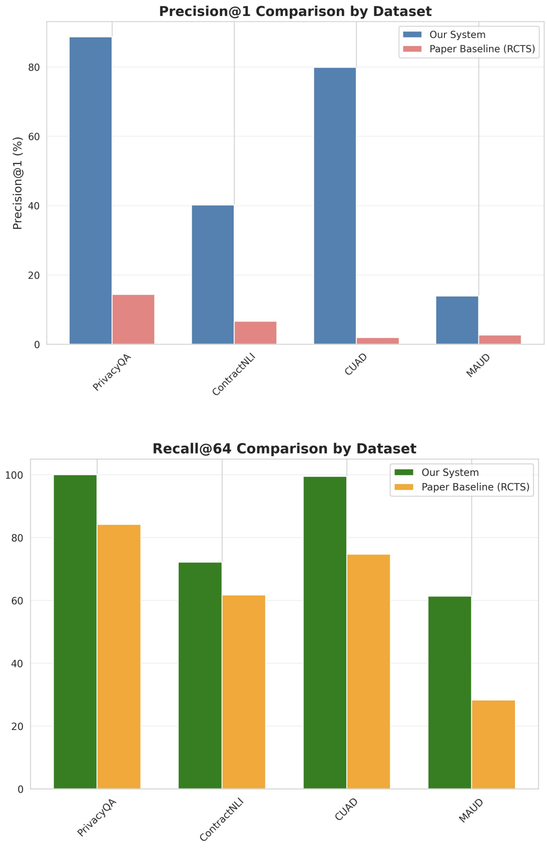

Figure 3: Comparison of our system with paper baseline

- **Massive Baseline Improvement:** The most encouraging result is our performance against the paper's baseline. The Baseline Comparison chart shows our system's 55.9% P@1 absolutely dominates the 6.4% baseline. We also see a 22-point jump in Recall@64 (84.4% vs 62.2%). This strongly validates that our embedding model (all-mpnet-base-v2) and indexing strategy are on the right track. [Fig.3]

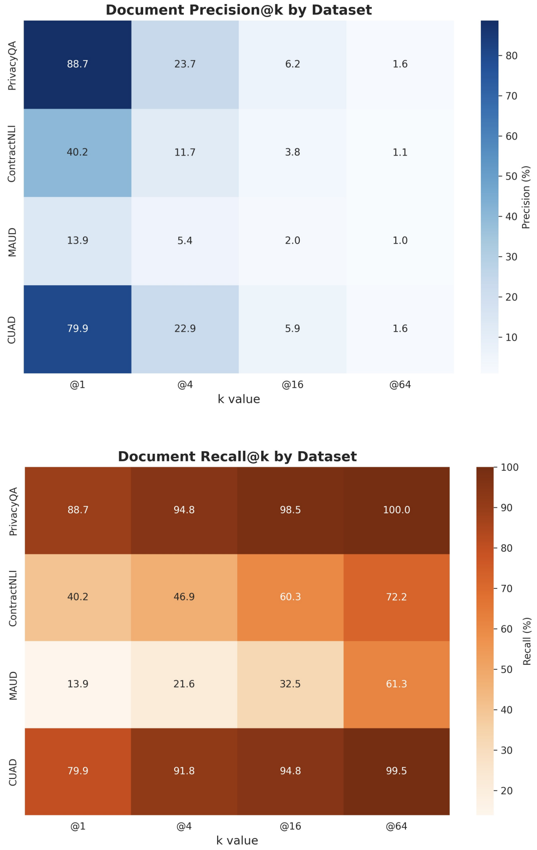

Figure 4: Heatmap of performance by legal domain

- **Performance by Legal Domain:** The Dataset Heatmap (Cell 20) provides a deeper look at our (strong) document-level performance. The results vary significantly by task:
  - **High Performance:** We performed very well on PrivacyQA (88.7% P@1) and CUAD (79.9% P@1).
  - **Low Performance:** We struggled significantly with MAUD (13.9% P@1).

This suggests our system is effective at specific fact and clause retrieval (PrivacyQA, CUAD) but finds the more nuanced, subjective tasks in MAUD (identifying agreement/disagreement) much more difficult. [Fig.4]

REFERENCES

[1] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *Proceedings of ICLR*, 2024.

[2] C. Chen, K. Liu, Z. Chen, Y. Gu, Y. Wu, M. Tao, Z. Fu, and J. Ye. INSIDE: LLMs' Internal States Retain the Power of Hallucination Detection. *arXiv preprint* arXiv:2402.03744, 2024.

[3] W. Su, C. Wang, Q. Ai, Y. Hu, Z. Wu, Y. Zhou, and Y. Liu. Unsupervised Real-Time Hallucination Detection based on Internal States of Large Language Models. In *Findings of ACL*, 2024.

[4] Z. Ji, D. Chen, E. Ishii, S. Cahyawijaya, Y. Bang, B. Wilie, and P. Fung. LLM Internal States Reveal Hallucination Risk Faced With a Query. *arXiv preprint* arXiv:2407.03282, 2024.

[5] M. Beigi, Y. Shen, R. Yang, Z. Lin, Q. Wang, A. Mohan, J. He, M. Jin, C. Lu, and L. Huang. InternalInspector (I^2): Robust Confidence Estimation in LLMs through Internal States. *arXiv preprint* arXiv:2406.12053, 2024.

[6] S. Dhuliawala, A. Asai, and H. Hajishirzi. Chain-of-Verification Reduces Hallucination in Large Language Models. In *Proceedings of ACL*, 2024.

[7] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, M. Kučerová, and V. Stoyanov. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Proceedings of NeurIPS*, 2020.

[8] M. Jeong, H. Park, D. Kim, S. Cho, and S. Lee. Self-BioRAG: Improving Biomedical Reasoning through Self-Reflective Retrieval-Augmented Generation. *Bioinformatics*, 2024.

[9] Stanford Institute for Human-Centered Artificial Intelligence (HAI). Generative AI and the Law: Risks of Hallucination and Misinformation. Stanford University Report, 2023. Available at: https://hai.stanford.edu/news

[10] J. Kim, S. Oh, and H. Lee. Lawyer LLaMA: Retrieval-Augmented Legal Reasoning with Self-Verification. *Applied Sciences*, MDPI, 2024.