Retrieval Augmented Generation for Legal Journals

Sheeba Moghal, Shingai Nindi, Powell Sheagren

Table of contents

Introduction

In a now infamous filing last June, lawyers representing a man who claimed to have suffered physical damages during a plane flight submitted a claim in part generated by AI to court. The claim used pre-existing cases to show precedent and all together seemed to have a coherent argument. There was only one problem, the judge and opposition couldn't find the cases referenced and, even after the lawyers followed up with ChatGPT, they were unable to prove their existence or validity. They claimed to be using the AI tool as a search engine and it cost them the case and some hefty fines for negligence. In this project, we hope to provide an alternative to this legal malpractice by using a retrieval augmented generation (RAG) model to build on a pre-existing LLM such that accurate and correct information can be found through searches. Law is just one of the many fields in which LLM 'hallucations' can cause tangible harm and lead to negative outcomes for users. The LLMs never claimed to be able to be experts on all topics but they do offer a framework for others to build specialized systems to retrieve specific information rather than simply predict the next word in a series. Since so much of legal work is research into precedent, we thought it would be an apt place to apply RAG so that a future model can save lawyers time while not costing them a drop in accuracy.

First, we will discuss the data that we sourced to complete this task, then we will discuss the methodology behind our models and the tools we used. Afterwards, we will show the results of our models and test their validity. Finally, we will return to the initial problem of researching legal cases and show the efficacy of the method to complete that task.

Methodology

Data Sources

For this project, the availability of pre-existing repositories of legal filings and documents significantly facilitated data access. This accessibility is largely attributed to the legal profession's reliance on easily searchable records to identify relevant precedents efficiently. Among the various legal datasets available, the **Caselaw Access Project** (case.law), a comprehensive repository dedicated to making all published U.S. court decisions publicly accessible wa selected. To streamline the scope of our study and optimize computational resources, Alaska state cases were focused upon. This targeted selection allowed us to manage the dataset more effectively while still providing a meaningful use case.

The original dataset was adapted to align with the requirements of our models, ensuring seamless integration and enhancing user experience in the final implementation. While we initially considered incorporating multiple data sources to enrich the dataset, the complexities of merging disparate datasets proved impractical within the current scope of the project. However, we anticipate that future iterations of this work, supported by more advanced computational resources, could expand to include additional data sources for a more robust and comprehensive model.

Data Preparation and Preprocessing

Data preparation and preprocessing are critical to ensuring the dataset is structured, consistent, and optimized for use with Retrieval-Augmented Generation (RAG) models. The dataset, sourced from the Caselaw Access Project, consisted of JSON files containing metadata and textual information about legal cases, including fields such as case name, abbreviation, decision date, jurisdiction, and detailed opinions. The preprocessing pipeline began with metadata cleaning, where text fields were normalized by converting them to lowercase, removing unnecessary punctuation, and ensuring uniform formatting of dates in the YYYY-MM-DD format. This step was crucial for maintaining consistency across the dataset, especially for fields like case names and legal abbreviations, which are prone to variability. Following this, long legal opinions were segmented into smaller passages of 300 to 512 characters. This segmentation was necessary to fit within the input size constraints of embedding models like BERT and MiniLM while preserving the contextual integrity of the text. Additional preprocessing involved deduplication to eliminate redundant records and tokenization to prepare the data for embedding generation. Special attention was given to aligning the metadata structure with the requirements of the retrieval systems, ensuring seamless compatibility for indexing and querying. By standardizing the dataset and breaking it into manageable segments, the preprocessing stage set the foundation for effective retrieval and summarization, reducing noise and ensuring high-quality inputs to the models.

Methods

RAG Framrwork

The Retrieval-Augmented Generation (RAG) framework was chosen for its ability to combine the strengths of information retrieval and natural language generation. RAG integrates a retriever to locate relevant documents or passages and a generator to synthesize coherent outputs conditioned on the retrieved content. This approach is particularly effective for legal data, where factual accuracy and context are paramount. Legal research often involves finding precedents or references that align closely with specific queries. By grounding the generative process in retrieved, verifiable legal documents, RAG minimizes the risk of hallucinations commonly associated with large language models (LLMs). This ensures that outputs are both contextually relevant and factually correct, making RAG an ideal choice for applications in the legal domain.

ColBERT: Contextualized Late Interaction over BERT

ColBERT (Contextualized Late Interaction over BERT) is a dense retrieval model designed for fine-grained semantic matching, ideal for legal frameworks where precision is critical. For a query $ Q $ and a document $ D $, ColBERT generates token-level embeddings:

$$Q = \{q_1, q_2, \dots, q_m\}, \quad D = \{d_1, d_2, \dots, d_n\}, \quad q_i, d_j \in \mathbb{R}^d$$

The similarity score between query token $ q_i $ and document token $ d_j $ is computed as:

$$s(q_i, d_j) = \langle q_i, d_j \rangle = \sum_{k=1}^{d} q_{i,k} \cdot d_{j,k}$$

Using **MaxSim**, the maximum similarity across document tokens is selected:

$$\text{MaxSim}(q_i, D) = \max_{j=1,\dots,n} s(q_i, d_j)$$

The final relevance score is:

$$S(Q, D) = \sum_{i=1}^{m} \text{MaxSim}(q_i, D)$$

ColBERT captures legal nuances with token-level granularity, contextual understanding, and efficient retrieval, ensuring precise and relevant document matching.

FAISS: Facebook AI Similarity Search

FAISS (Facebook AI Similarity Search) is a scalable framework for similarity search and clustering of dense vector embeddings, making it highly suitable for legal frameworks. Legal documents are represented as high-dimensional embeddings $d_i \in \mathbb{R}^d$, and FAISS retrieves relevant documents by minimizing the Euclidean distance between query q and document embeddings:

$$\text{dist}(q, d_i) = \|q - d_i\|_2^2 = \sum_{j=1}^{d} (q_j - d_{i,j})^2$$

FAISS supports advanced indexing techniques like Inverted File (IVF) and Hierarchical Navigable Small World (HNSW) graphs to cluster embeddings, reducing the search space and enhancing scalability. For a query, FAISS retrieves the top $ k $ most similar documents:

$$\text{Top-}k(q) = \text{argmin}_{D'} \, \{\text{dist}(q, d_i) \mid d_i \in D, \ |D'| = k\}$$

In legal frameworks, FAISS excels due to its scalability, speed, and ability to handle large datasets, ensuring precise, semantically relevant document retrieval for real-time legal research.

BART: Summarization for Legal Frameworks

BART (Bidirectional and Auto-Regressive Transformer) is a transformer-based encoder-decoder model designed for sequence-to-sequence tasks, making it ideal for legal document summarization. BART is pre-trained as a denoising autoencoder, which reconstructs corrupted text, and fine-tuned for abstractive summarization tasks. The encoder maps the input sequence $ X = {x__1, x__2, ..., x__n} $ to a latent representation, while the decoder generates the output sequence $ Y = {y__1, y__2, ..., y__m} $ by maximizing the conditional probability:

$$P(Y|X) = \prod_{t=1}^{m} P(y_t | y_{<t}, X)$$

In legal frameworks, BART is used to generate concise, coherent summaries of lengthy legal documents, ensuring the essence of the text is preserved. Its attention mechanism enables BART to focus on the most relevant parts of the document, making it adept at handling complex legal language. This ability to synthesize key information helps lawyers quickly access critical insights while maintaining high accuracy and readability.

Performance Metrics Used:

The evaluation of the legal document retrieval system uses multiple metrics to assess retrieval accuracy and ranking quality:

1. **Precision@k**
   Precision@k measures the proportion of relevant documents in the top k results:

$$\text{Precision@k} = \frac{\text{Number of Relevant Documents Retrieved in Top-}k}{k}$$

2. **Recall@k**
   Recall@k evaluates the fraction of all relevant documents that are retrieved within the top k results:

$$\text{Recall@k} = \frac{\text{Number of Relevant Documents Retrieved in Top-}k}{\text{Total Number of Relevant Documents}}$$

3. **F1-score@k**
   The F1-score@k combines precision and recall into a single harmonic mean:

   $$\text{F1-score@k} = 2 \cdot \frac{\text{Precision@k} \cdot \text{Recall@k}}{\text{Precision@k} + \text{Recall@k}}$$

4. **nDCG@k (Normalized Discounted Cumulative Gain)**
   nDCG@k evaluates the ranking quality of the retrieved documents, rewarding relevant documents appearing earlier in the list:

   $$\text{nDCG@k} = \frac{\text{DCG@k}}{\text{IDCG@k}}$$

   where DCG is the Discounted Cumulative Gain, and IDCG represents the Ideal DCG for perfect ranking.

5. **Mean Average Precision (MAP)**
   MAP aggregates the average precision across all queries, reflecting the system's overall retrieval performance.

These metrics collectively evaluate the system's ability to retrieve accurate, relevant, and well-ranked documents, providing insights into its effectiveness and highlighting areas for improvement.

Discussion

Model 1: FAISS + BART

The implemented code for legal case retrieval and querying establishes a robust pipeline integrating embedding generation, indexing, querying, and summarization, ensuring efficiency and scalability for legal research. Beyond preprocessing, which standardizes and organizes raw legal data, the system leverages advanced methods to handle large datasets effectively.

After preprocessing, legal texts are converted into dense vector embeddings using a retained pre-trained transformer model, such as `all-MiniLM-L6-v2` from the sentence-transformers library. This model generates high-quality semantic embeddings, ensuring meaningful representation of legal texts. To handle text length constraints, documents are segmented into manageable passages (300–512 characters), preserving context while optimizing compatibility with the embedding model.

The embeddings are indexed using FAISS (Facebook AI Similarity Search), a framework designed for fast and scalable similarity search. FAISS creates a searchable structure, such as a flat L2 index, allowing rapid retrieval of the most relevant documents. Each embedding is linked to metadata fields like case name, abbreviation, and decision date, with the index saved to disk for reusability. This setup ensures efficiency and scalability for querying extensive datasets.

The query system supports semantic and metadata-based searches. Semantic queries involve embedding user input with the same model used for the document embeddings, ensuring alignment for accurate similarity matching. The system retrieves and ranks the top results. Metadata-based searches use fuzzy matching to identify partial matches for fields like case name or decision date, offering flexibility for user needs.

To enhance usability, a BART (Bidirectional and Auto-Regressive Transformer) summarization model generates concise summaries of retrieved documents. These summaries distill essential information, reducing the time required to analyze lengthy legal cases.

By integrating FAISS for efficient indexing and BART for actionable summarization, the system provides a scalable, user-friendly, and accurate solution for legal case retrieval and analysis.

Model 2: Colbert + BART

The implemented legal case retrieval system integrates ColBERT, DistilBERT-based embeddings, and BART to deliver an efficient and accurate solution for querying and summarizing legal cases. The cleaned queries, based on the input are embedded into dense vector representations using DistilBERT, a lightweight transformer model known for its efficiency and contextual understanding. DistilBERT produces compact embeddings while retaining semantic richness, making it suitable for large-scale legal datasets.

The generated query embeddings are compared to pre-indexed document embeddings using ColBERT's token-level similarity computation. The ColBERT framework allows for fine-grained, token-by-token interaction between the query and document embeddings. Instead of relying solely on overall document embeddings, ColBERT calculates relevance at the token level, ensuring that even nuanced legal terms or phrases in the query are matched to corresponding tokens in the documents. This process retrieves the most contextually relevant legal cases based on the user's query type, whether it involves case names, abbreviations, decision dates, or free-text queries. The retrieved results are ranked by similarity, ensuring that the most relevant cases appear at the top.

Once the retrieval step is complete, the retrieved legal cases undergo summarization using the BART (Bidirectional and Auto-Regressive Transformer) model. BART, fine-tuned for abstractive summarization, processes the textual content of the retrieved cases by encoding the context and generating a coherent summary. This summarization reduces the cognitive load for users by presenting the key details and essential insights from lengthy and complex legal texts in a concise format.

The integration of ColBERT's advanced retrieval logic with DistilBERT embeddings ensures precise semantic alignment, while BART summarization enhances usability by distilling relevant information into actionable insights.

Interactive and Advanced Query System

The interactive query system is designed to handle multiple query types, including case names, abbreviations, decision dates, jurisdictions, and custom legal queries, while ensuring precise and partial matches. User queries are preprocessed to enhance alignment with indexed document metadata. Preprocessing begins with text normalization, where queries are converted to lowercase, punctuation is removed, and whitespace is standardized. Legal terms such as "v." and "vs." are normalized for consistency. Additionally, filler words like "What about" or "Can you find" are stripped from the query, leaving only the meaningful text. For metadata-based searches, fuzzy matching techniques are employed, allowing partial matches between the user query and fields such as case names or abbreviations. This ensures flexibility in handling queries where users may not input exact terms.

For custom legal queries, the system dynamically embeds the input text using the DistilBERT model, which transforms the query into a dense vector representation capturing its semantic context. This embedding is compared with pre-indexed document embeddings using ColBERT's token-level similarity computation. The token-level interaction ensures that the system can identify documents relevant to complex, open-ended queries by aligning query tokens with corresponding document tokens. For example, a query like "precedents for unlawful restraint" retrieves documents not only containing those words but also semantically related cases.

Once relevant cases are retrieved, they are ranked by similarity and displayed with associated metadata such as case name, decision date, and jurisdiction. Users are provided with options to view snippets of the text or generate summaries. The summarization is handled by the BART model, which abstracts the retrieved documents into concise, coherent summaries. This interactive workflow combines robust preprocessing, advanced embedding-based retrieval, and summarization to ensure that users can access precise, contextual, and actionable information efficiently, regardless of the query's complexity or specificity.

Results

Model Comparison and Evaluation

The performance of both models was assessed using metrics such as Precision@k, Recall@k, F1-score@k, and normalized discounted cumulative gain (nDCG@k), focusing on the top *k* results to measure retrieval accuracy and ranking quality. Two prompts, "What about Hillyer" and "Case on McIntosh," were used to compare the models. The results are detailed below.

Results for "What about Hillyer"

Model	k	Precision	Recall	F1-Score	nDCG	Match Type
ColBERT	1	1.0	1.0	1.0	1.0	Partial
ColBERT	3	0.33	1.0	0.50	1.0	Partial
ColBERT	5	0.20	1.0	0.33	1.0	Partial
FAISS	1	1.0	1.0	1.0	1.0	Partial
FAISS	3	0.33	1.0	0.50	1.0	Partial
FAISS	5	0.20	1.0	0.33	1.0	Partial

Results for "Case on McIntosh"

Model	k	Precision	Recall	F1-Score	nDCG	Match Type
ColBERT	1	1.0	0.50	0.67	1.0	Partial
ColBERT	3	0.67	1.0	0.80	1.0	Partial
ColBERT	5	0.40	1.0	0.57	1.0	Partial
FAISS	1	1.0	0.33	0.50	1.0	Partial
FAISS	3	0.33	0.33	0.33	0.80	Partial
FAISS	5	0.20	0.33	0.25	0.70	Partial

Analysis of Results

The results show that **ColBERT consistently outperforms FAISS** on non-specific prompts across all metrics, particularly in recall, F1-score, and nDCG. This difference stems from ColBERT's **token-level interaction mechanism**, which provides fine-grained matching between queries and documents. By preserving token-level granularity, ColBERT captures contextual nuances in legal text better than FAISS, which relies on pre-computed dense vector similarities. This makes ColBERT especially effective for handling complex or ambiguous legal queries.

FAISS, while efficient for large-scale similarity searches, lacks the ability to dynamically adjust token-level relevance, leading to lower recall and F1-scores in cases where subtle legal contexts are critical. Additionally, FAISS requires re-indexing when new data is added, whereas ColBERT scales incrementally, supporting more flexible use in dynamic legal datasets.

Summarization with BART

Both models use the **BART summarization model** for generating concise, coherent summaries of retrieved documents. This decision ensures consistency in the summarization step, allowing for a direct comparison of retrieval methods. By isolating the retrievers as the only differentiating factor, the evaluation highlights the impact of retrieval techniques on system performance. BART's encoder-decoder architecture is equally effective regardless of the retriever, ensuring that the summaries are contextually accurate for both FAISS and ColBERT results.

Model Limitations

The system has several risks and limitations that could impact its effectiveness. The BART summarization model, while powerful, is constrained by its input context window, which may truncate lengthy legal documents and omit critical details, potentially affecting the completeness of summaries. The reliance on pre-trained models like DistilBERT and ColBERT is another limitation, as these models are trained on general-purpose datasets and may not fully capture the nuances of legal language or domain-specific terms. Additionally, the FAISS model requires re-indexing whenever new data is added, limiting its scalability in dynamic datasets compared to ColBERT, which supports incremental updates. The system's dependence on high-quality metadata and preprocessing also poses a risk, as incomplete or inconsistent data could negatively affect retrieval accuracy and ranking. Lastly, both retrieval methods are computationally intensive, with ColBERT being particularly resource-heavy due to token-level interaction, which could limit real-time application in large-scale systems.

Conclusion

As demonstrated, AI can be effectively utilized to streamline the legal research process, particularly in citing accurate sources and identifying those relevant to specific cases. This study aimed to address challenges in legal document retrieval, detailing the methodology and systems employed, followed by an evaluation of the two models developed. Both models performed well in extracting relevant information from the documents, and despite stylistic differences, the final outputs were found to be largely similar in terms of accuracy and relevance.

Revisiting the initial scenario, the tool developed would assist in retrieving sources that are grounded in reality rather than hallucinated by large language models. However, it does not address other aspects of malpractice, such as verification or legal interpretation. Future developments could integrate retrieval-augmented generation with complete legal filings to close this gap. As machine learning techniques continue to evolve, it becomes increasingly critical to implement mechanisms at every stage to ensure the quality and accuracy of outputs. Without such safeguards, the potential for errors or misinformation could expose AI developers and legal practitioners to liability, underscoring the need for careful oversight and ethical implementation in legal contexts.

References

1. Caselaw Access Project. (2024). Retrieved [29th November, 2024], from [url].