VII Semester Project Work-1

Seminar Report

On

NOMOSAI: LEGAL AI ASSISTANT

Submitted by

RANIA MEHREEN FAROOQ (1604-21-733-002)

MAHEEN ILYAS (1604-21-733-004)

Project Guide: FAHMINA TARANUM

Professor & Associate Head

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY

2024-2025

1. ABSTRACT

The legal field is known for its complexity due to multi-level hierarchies, domain-specific vocabulary, and nuanced interpretations. Traditional judiciary systems often rely on human decision-making and precedents, making the process resource-intensive and time-consuming. Additionally, access to affordable legal assistance remains a significant barrier for many individuals. As an emerging field, the application of LLMs in the legal field is still in its early stages, with multiple challenges that need to be addressed. In the Indian legal system, the lack of sufficient datasets and open-access legal documents poses additional barriers to developing reliable AI solutions. Ethical considerations, including bias, accountability, and transparency, further complicate the deployment of AI in this sensitive domain. Our objective is to develop Legal AI Assistant "NomosAI" designed to address the challenges faced in the legal domain by leveraging advanced Artificial Intelligence (AI) technologies, particularly Retrieval-Augmented Generation (RAG). This project aims to simplify legal research and document analysis through a user-friendly platform that integrates AI models fine-tuned for legal tasks. The system employs state-of-the-art tools, including **LangChain**, **Hugging Face Embeddings**, **Recursive Character Text Splitter** and **ChromaDB,** to manage document ingestion, text splitting and vector storage. A high-performing large language model **(Llama 3.3)** further ensures accurate and context-aware responses. NomosAI allows users to upload case files, query constitutional provisions, and interactively analyze legal documents through an intuitive interface built on **Next.js**, backed by a **FastAPI** backend.The solution addresses critical challenges in legal information retrieval, offering a scalable, efficient and user-friendly platform for legal professionals, researchers and the broader legal community.

LIST OF FIGURES

2. INDEX

3. INTRODUCTION

1. Basic Background

a. The importance of the field in which the work is being done

Since November 2022, with the launch of ChatGPT, and especially after GPT4, there has been an exponential increase in the use of LLMs in various research fields, including code generation, economics, healthcare, and education. Amidst this surge of innovation, the legal domain has remained particularly challenging. Legal systems are hard to understand and explain because of their complicated structure, specialised language, and varying interpretations. This complexity makes it difficult for both the general public and professionals to navigate the legal field. Although technological advancements have the potential to create LLMs that simplify these systems, their effectiveness has not yet been proven, and multiple challenges remain.

b. Existing Issues and Current Trends

The multi-level hierarchies, domain-specific vocabulary, and nuanced interpretations inherent to legal matters pose significant challenges for these models. Consequently, the outputs generated by the models do not consistently provide the necessary depth and precision to provide meaningful assistance in real-world legal scenarios. Off-the-shelf LLMs struggle to fully capture the complexities and nuances of legal language and reasoning. They may fumble with the specialised terminology and citation formats or outright hallucinate domain-specific knowledge, losing the rigour and precision essential in legal contexts.

c. Potential for work in the area

To truly leverage the capabilities of LLMs in the legal field, it is necessary to develop models that are fine-tuned and adapted to the specific requirements of legal research and practice. This involves training LLMs on extensive collections of legal texts, incorporating domain-specific knowledge, and optimising them for tasks such as legal document retrieval, summarisation, and analysis. Specialised legal LLMs can better understand the context and

meaning of legal language, handle the unique structures and formats of legal documents, and provide more accurate and relevant results.

2. Problem Statement

The traditional judiciary system possesses several distinct characteristics that are essential to comprehend when considering the application of judicial artificial intelligence. These characteristics encompass a reliance on human decision-making, a lack of flexibility, and substantial resource consumption.

One of the primary features of the traditional judiciary is its dependence on human decision-making, particularly that of judges, prosecutors, and lawyers. Throughout the process of reasoning and evidence collection, legal professionals often refer to case-specific circumstances, legal provisions, and precedents, in conjunction with their professional knowledge, to formulate judgements and decisions. The final judgement or defence is then presented through a trial.

Another key aspect of the traditional judiciary is its reliance on precedents during the decision-making process. Previous judgements in similar cases and relevant legal provisions often guide the decisions of courts. In many judicial systems, the judgements of the highest court are considered authoritative and binding, serving as a reference for other courts in relevant cases.

The traditional judiciary is often time- and resource-consuming, particularly when dealing with a large number of cases. The process of case hearings, summoning witnesses, and collecting evidence can prolong the trial process and consume significant resources. This can lead to situations where there are many cases but limited personnel, resulting in backlogs and delays.

Furthermore, many individuals are likely to face a legal dispute at some point in their lives, but their lack of understanding of how to navigate these complex issues often renders them vulnerable. The advancement of natural language processing opens new avenues for bridging this legal literacy gap through the development of automated legal aid systems. Expert legal assistance is often prohibitively expensive, which results in a considerable number of vulnerable individuals being left unprotected or exploited due to their inability to afford it. This barrier to accessing legal information

fosters a significant imbalance within the legal system, impeding the universal right to equal access to justice for all.

3. Objective

The legal AI tool Nomos AI presented in this paper aims to revolutionise the way law professionals and law students conduct research by utilising advanced AI technologies like Retrieval-Augmented Generation (RAG).

Purpose: To streamline summarization, retrieval and semantic matching

Summarisation

Condensing large volumes of legal text into concise representations without losing critical details. This is crucial in the legal domain, where documents like case law, statutes and contracts can be lengthy and complex.

Example: For a lengthy case judgement, the assistant might summarise the key decision, the involved parties and the legal principles applied.

Retrieval

Searching through vast legal databases and pinpointing documents that match the search criteria.

Example: A query like "cases related to breach of contract in India" retrieves judgements, statutes and articles addressing the issue.

Semantic Matching

Semantic matching evaluates the meaning and context of user queries and legal texts, rather than relying solely on keyword matching

Example Query: "Car accident cases"

How Semantic Matching Works: Recognises related terms like "vehicle collision" or "motor accident" even if they are not explicitly mentioned in the query.

Legal Education: Enabling users to ask queries related to the constitution, laws, and regulations to stay up-to-date with the amendments.

4. Organization of Report

The rest of this work is organised as follows: First, we take a brief introduction to related works, including foundation models, benchmark tests for legal AI models, the datasets, and the literature review on the work done so far in this domain. Second, we

reveal the details of the system analysis, which include the existing problem, the proposed solution, and the technical specifications. Third, we design the system and present the architecture diagram. Fourth, we will discuss the plan of work and the project timeline. Finally, we conclude our work and provide the reference.

4. LITERATURE SURVEY

1. **LegalBench-RAG: A Benchmark for Retrieval-Augmented Generation in the Legal Domain**

    a. **Existing Benchmarks:** Existing benchmarks like LegalBench do not evaluate the retrieval quality over a large corpus, which is crucial for RAG-based systems in legal. This limitation highlights the need for a benchmark that can specifically assess *retrieval mechanisms* in legal applications.

    b. **A Benchmarking Dataset: LegalBench-RAG**

    Starting Point: LEGALBENCH: LegalBench is a collaboratively constructed legal reasoning benchmark consisting of 162 tasks covering six different types of legal reasoning. However, Legal Bench does not benchmark the ability to extract the correct context from within a larger corpus. This limitation inspired the LegalBench-RAG. Four datasets were constructed for retrieval benchmark: Privacy Question Answering (PrivacyQA), Contract Understanding Atticus Dataset (CUAD), Mergers and Acquisitions Understanding Dataset (MAUD), Contract Natural Language Inference (ContractNLI).

    Tracing Back to Original Sources: To transform LegalBench into a retrieval benchmark, a comprehensive process was undertaken to trace each text segment used in LegalBench, back to its original location within the source corpus. This ensured that the benchmark accurately reflects the retrieval capabilities needed to locate exactly the relevant information within a large legal corpus.

    Construction Process: Each of the four source datasets were converted into LegalBench-RAG queries. Legal Bench will take a given query, and convert the annotation label from the CUAD dataset, into a Yes/No label based on whether or not the label matches the query. LegalBench-RAG on the other hand, will ask the same query and the label will be the filename and span of the relevant text necessary to answer the query.

c. **Results:** The performance of all methods across the four datasets was analysed and as expected, the PrivacyQA dataset is the easiest benchmark. It consistently yielded the highest scores across all methods, indicating that this dataset was easier for the models to interpret.

2. **Interpretable Long-Form Legal Question Answering with Retrieval-Augmented Large Language Models**

This work proposes an end-to-end methodology designed to generate long-form answers to any statutory law questions, utilizing a "retrieve-then-read" pipeline. For instance, legal text summarization holds the potential to simplify complex legal documents for the layperson, while legal judgment prediction could unveil insightful correlations between an individual's situation, and the probable legal outcome. Similarly, legal question answering (LQA) could offer affordable, expert-like assistance to the masses, thereby empowering marginalized parties when utilized for public welfare.

The main contributions of this paper are:

1. A novel dataset for long-form question answering (LFQA) in the legal domain and French language, comprising 1,868 legal questions, meticulously annotated by legal professionals, with detailed answers and references to relevant legal provisions, drawn from a substantial knowledge corpus containing 27,942 statutory articles.
2. A comprehensive evaluation of the retrieve-then-read framework in the context of legal LFQA, while emphasizing interpretability and exploring various learning strategies for the reader.

Dataset: Long-form Legal Question Answering (LLeQA): A significant contribution of this work is the introduction of the Long-form Legal Question Answering (LLeQA) dataset, which comprises 1,868 expert-annotated legal questions in French. This dataset not only includes detailed answers but also references pertinent legal provisions, making it a valuable resource for training and evaluating LQA systems. The authors detail their rigorous process for dataset construction, which involved collaboration with legal professionals and extensive curation of statutory articles.

The proposed methodology consists of two main components: a retriever and a reader. The retriever employs a bi-encoder model to efficiently fetch relevant legislative articles from a large corpus, while the reader utilizes an instruction-tuned large language model (LLM) to formulate detailed answers based on the retrieved information. This approach allows for the generation of comprehensive and interpretable responses, moving beyond the limitations of existing LQA systems that often provide brief or uninformative answers. The paper discusses the limitations of traditional evaluation metrics like ROUGE for assessing long-form answers. The authors emphasize the need for more robust evaluation methods that can accurately reflect the quality of generated responses, particularly in legal contexts where precision is paramount. In conclusion, the "retrieve-then-read" framework shows promise but has limitations. LLeQA can advance interpretable legal question answering and democratize access. Future work should address hallucinations and improve evaluation metrics. Ethical considerations are crucial to prevent misuse of LQA systems.

3. **Athena: Retrieval-augmented Legal Judgment Prediction with Large Language Models**

The paper "Athena: Retrieval-Augmented Legal Judgment Prediction with Large Language Models" by Xiao Peng and Liang Chen explores the integration of Retrieval-Augmented Generation (RAG) within large language models (LLMs) to enhance legal judgment prediction (LJP). The authors argue that while LLMs have shown promise in various applications, their performance in specialized legal tasks remains underexplored, particularly concerning the dynamic nature of legal regulations and judgments. The authors note that traditional methods for LJP often fail to adapt to new legal precedents or regulations, leading to limitations in real-world applicability.

Methodology: The proposed framework, Athena, consists of two primary workflows: prompting and knowledge retrieval. The prompting workflow involves analyzing legal cases and generating structured outputs, while the knowledge retrieval workflow utilizes a semantic retrieval mechanism to fetch relevant accusations from a constructed knowledge base. By embedding external knowledge into the LLM's

inference process, Athena aims to mitigate issues such as hallucinations and inaccuracies commonly associated with LLMs.

Knowledge Base Construction: A significant aspect of Athena's methodology is the construction of a knowledge base that includes various legal accusations along with their descriptions and examples. This knowledge base is built through a three-step process: deduplication of accusation labels, query rewriting using LLMs, and vectorization for semantic similarity searching. This structured approach allows for efficient retrieval of relevant information during the prediction process.

Experimental Results: The authors conduct experiments using the CAIL2018 dataset, demonstrating that Athena achieves state-of-the-art results in LJP tasks. Their findings indicate that as the model's capacity increases—from GPT-3.5 to GPT-4—the accuracy of predictions also improves significantly. The results underscore the effectiveness of RAG in enhancing LLM performance by providing contextually relevant information.

4. **CBR-RAG: Case-Based reasoning for Retrieval-Augmented Generation in LLMs for Legal Question Answering**

The paper "CBR-RAG: Case-Based Reasoning for Retrieval-Augmented Generation in LLMs for Legal Question Answering" presents an innovative approach to enhance legal question answering (QA) systems through the integration of Case-Based Reasoning (CBR) and Retrieval-Augmented Generation (RAG) methodologies. The authors argue that combining these techniques can significantly improve the accuracy and contextual relevance of responses generated by large language models (LLMs), particularly in knowledge-intensive tasks such as legal inquiries. The motivation behind this research lies in leveraging CBR to provide a structured retrieval mechanism that enhances the context available to LLMs, thereby improving the quality of generated answers.

Methodology: The proposed CBR-RAG framework integrates CBR's initial retrieval stage with RAG's generative capabilities. This approach involves several key components:

- **Case Base Construction:** The authors utilize the Australian Open Legal QA (ALQA) dataset, which includes 2,124 LLM-generated question-answer pairs

supported by textual snippets from legal documents. This dataset serves as the foundation for building a case base that facilitates effective retrieval of relevant legal precedents.

- **Embedding Representations:** The paper introduces dual embedding forms—intra-embeddings for attribute matching and inter-embeddings for information retrieval. This dual approach allows for nuanced comparisons between questions, supporting texts and entities, enhancing the model's ability to retrieve contextually relevant cases.
- **Retrieval Strategies:** The authors outline three comparison strategies for case retrieval: intra-, inter-, and hybrid-embedding based retrieval. These strategies enable the model to match queries with cases based on different similarity metrics, thereby optimizing the retrieval process.

Experimental Results: The evaluation of the CBR-RAG framework demonstrates significant improvements in the quality of generated answers compared to traditional methods. By incorporating context from retrieved cases, the model achieves higher accuracy and relevance in its responses. The findings suggest that CBR's case reuse mechanism effectively reinforces similarities between user queries and the evidence base, leading to more informed outputs.

5. **CLERC: A Dataset for Legal Case Retrieval and Retrieval-Augmented Analysis Generation**

The paper titled "CLERC: A Dataset for Legal Case Retrieval and Retrieval-Augmented Analysis Generation" by Hou et al. presents a significant contribution to the intersection of legal studies and artificial intelligence (AI). The authors address the challenges faced by legal professionals in drafting analyses that rely on accurate citations of precedents. They introduce the CLERC dataset, designed to enhance information retrieval (IR) and retrieval-augmented generation (RAG) tasks within the legal domain. This dataset aims to bridge the gap between computational efficiency and the practical needs of legal professionals, providing a robust resource for training AI models.

Dataset Construction: The CLERC dataset is constructed from the Caselaw Access Project (CAP), encompassing over 1.84 million federal case documents. The authors

detail their methodology for transforming this corpus into a structured dataset suitable for both IR and RAG tasks. Key features include:

- **Document Types:** The dataset includes long case documents (CLERC/doc) and chunked passages for retrieval (CLERC/passage).
- **Citations:** With over 20.7 million total citations, the dataset provides a rich resource for evaluating citation retrieval accuracy.
- **Query Design:** Queries are formulated by removing central citations from legal texts, simulating real-world scenarios where lawyers need to find supporting cases.

Evaluation of Models: The authors benchmark state-of-the-art models against their dataset, revealing that while models like GPT-4o achieve high ROUGE F-scores in generating analyses, they also exhibit significant hallucination issues. Zero-shot IR models demonstrate limited effectiveness, achieving only 48.3% recall at 1000 citations. This evaluation underscores the challenges that remain in developing reliable AI systems for legal applications.

Long-Context Generation and Retrieval: The paper discusses the constraints of large language models (LLMs) regarding context length, which is critical for processing lengthy legal texts. The authors explore various strategies to extend context capabilities, including efficient attention mechanisms and RAG methodologies. They also note that while there has been progress in long-context retrieval models, comprehensive benchmarks specifically tailored to legal tasks are still lacking.

6. **To What Extent Have LLMs Reshaped the Legal Domain So Far? A Scoping Literature Review**

The objective of this paper is to provide a comprehensive survey of legal LLMs, not only reviewing the models themselves but also analysing their applications within the legal systems in different geographies. Two databases (i.e., SCOPUS and Web of Science) were considered alongside additional related studies.

This scoping review seeks to explore five primary inquiries regarding the application of LLMs within the legal field, specifically focusing on their operational use cases and the methodologies employed in their construction:

- **RQ1:** Which LLM tools are considered leading in the field, and which are best suited for legal applications according to the current open-access state-of-the-art research?
- **RQ2:** What are the primary sources for data extraction and the best strategies for dataset development within the legal domain?
- **RQ3:** What are the challenges of LLMs in addressing legal tasks?
- **RQ4:** What are the main strategies for increasing the performance of LLMs in addressing legal tasks?
- **RQ5:** What are the main limitations of current LLMs for the legal domain?

Methodology

The study considered the PRISMA extension for Scoping Reviews (PRISMA-ScR) and PRISMA2020 methodology to provide syntheses of the state of knowledge in legal applications based on artificial intelligence. Next, the distribution and trends within the selected body of literature were analysed. A statistical analysis was conducted that focused on three key aspects: geography, publication month, and publication channel.

1. **Geography**

   The geographical distribution of the articles was analysed to identify the regions most active in research on the legal applications of LLMs. This geographic analysis helped pinpoint global centres of research and innovation within the domain. The US + CA was the geography with the highest number of studies conducted, with Europe and China standing out as second and third in these research efforts.
2. Additionally, the number of articles published per month was examined to uncover any temporal trends or spikes in research activity, providing insights into the evolution of interest and focus over time. A clear increase in the number of publications concerning LLMs used in the legal domain from 2022 to 2024.
3. Lastly, the distribution of articles across various publication channels was analysed, including journals, conferences, and other platforms, to better understand the preferred avenues for disseminating advancements in this rapidly evolving field. A significant portion of studies were available on platforms like Arxiv, which are preprint repositories allowing researchers to

share their work quickly before formal peer review. This pattern highlights the balance between the need for rapid dissemination through preprint servers and the importance of publishing in established academic journals.

7. **Attention is all you need**

The paper "Attention Is All You Need" introduces the Transformer model, a novel architecture that relies solely on attention mechanisms, eschewing the recurrent and convolutional layers traditionally used in sequence transduction tasks. Prior to the introduction of the Transformer, sequence modeling predominantly utilized Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). These models, while effective, faced limitations due to their sequential nature, which hindered parallelization and increased training times. The authors highlight that attention mechanisms had begun to show promise in improving model performance by allowing for the modeling of dependencies irrespective of their distance within input or output sequences. However, these mechanisms were typically integrated into RNN frameworks rather than being employed as standalone architectures.

The Transformer architecture fundamentally changes how sequence data is processed by introducing a purely attention-based mechanism. The model consists of an encoder-decoder structure where both components are built using stacked self-attention layers and feed-forward networks.

The authors conducted experiments on machine translation tasks, specifically focusing on English-to-German and English-to-French translations. The Transformer achieved state-of-the-art results on both tasks. These results not only surpassed previous benchmarks but also demonstrated significantly reduced training times—achieving these scores after just 3.5 days on eight GPUs compared to weeks for earlier models.

Despite its successes, the Transformer model is not without challenges:

- **Computational Resources:** While it is more efficient than RNNs in terms of training time, it still requires substantial computational power and memory.
- **Data Requirements:** Transformers often require large datasets for effective training, which can be a barrier in low-resource settings.

5. SYSTEM ANALYSIS

1. Existing System

Overall, the selected literature revealed several key tasks for applying LLMs within the legal domain.

1. Legal case retrieval is a key task in case law systems, making this task particularly challenging and vital for developing intelligent legal systems. Given the complexity of legal texts and the importance of precedents, this task is both difficult and essential.
2. Legal question answering involves training AI to provide accurate and interpretable answers to legal queries. This area of research focuses on bridging the gap between complex legal information and the general public by offering automated solutions that can deliver expertlike guidance.
3. Document drafting is another important legal task where LLMs are making significant contributions, particularly in the creation of legal documents such as contracts and reports.
4. Semantic annotation of legal texts uses LLMs to label parts of legal documents, such as identifying rhetorical roles or clause types.

Challenges in the Indian Legal System

Despite the promising applications of RAG and LLMs, the literature indicates several challenges specific to implementing these technologies in the Indian legal system:

1. **Lack of Sufficient Datasets:** A significant barrier is the scarcity of comprehensive, publicly accessible legal datasets that reflect the unique structure and terminology of Indian law. Most existing benchmarks focus on Western legal systems, which may not adequately capture the nuances required for effective application in India.
2. **Open Access Legal Documents:** The limited availability of open-access legal documents further complicates efforts to train and evaluate RAG systems tailored to Indian law. Without sufficient data, it becomes challenging to adapt existing models or develop new ones that can effectively address local legal queries.

3. **Ethical and Practical Considerations:** As with any AI application in sensitive domains like law, ethical considerations regarding bias, accountability, and transparency are paramount. Future research must explore these dimensions to ensure that AI tools are used responsibly within the legal framework.

2. Proposed System

This project addresses challenges like hallucinations, contextual irrelevance, restricted dataset scopes and retrieval biases, delivering a solution that is accurate, scalable and contextually adaptable for diverse legal scenarios. NomosAI introduces a simplified yet effective website that integrates advanced AI tools, practical legal-specific adjustments and clear mechanisms to improve interpretability.

Key Components

- The frontend is designed to provide an intuitive and user-friendly interface for users to upload case files, query legal documents and view results. Built using Next.js and styled with Tailwind CSS, it features a streamlined document upload system, a search bar for queries, and a responsive display for retrieved results and summaries. Clear formatting ensures that users can easily understand legal citations, context and conclusions.
- The backend, powered by FastAPI, manages all data processing, model interactions and system. It serves as the bridge between the user input and the RAG pipeline, handling:
    - Document uploads and preprocessing.
    - Query forwarding to the RAG pipeline.
    - Response formatting and displaying it on the frontend.
- The RAG pipeline is the core of the system, responsible for retrieving relevant legal information and generating accurate, interpretable responses. It consists of the following components:
    1. **Document Processing**

       The document processing module begins by extracting and cleaning text from uploaded legal documents, ensuring that the content is ready for downstream processing. A recursive character text splitter is used to break down large documents into smaller, manageable chunks while

maintaining semantic coherence. These divisions are made along meaningful boundaries, such as legal clauses, paragraphs or sections to preserve context. Once split, the text chunks are transformed into vector embeddings using specialized Hugging Face models trained on legal corpora. These embeddings are then stored in ChromaDB, a scalable and efficient vector database designed to support high-speed retrieval.

2. **Query Handling and Retrieval**
   User queries are processed by converting them into vectorized embeddings that can be matched against the stored document embeddings. This similarity matching mechanism retrieves the most relevant chunks of text, ensuring that the returned information aligns closely with the query context.

3. **Response Generation and Verification**
   Once the relevant text chunks are retrieved, Llama 3.3 is used to generate concise and contextually accurate responses. The ROGUE metrics are used to check the recall and precision of the response generated by the LLM.

The proposed solution addresses the key limitations of existing RAG systems in the legal domain by integrating advanced retrieval and summarization techniques with domain-specific optimizations. With its focus on accuracy, interpretability and global applicability, this system represents a significant advancement in legal research and decision-making technology.

3. Technical Specifications

- **Frontend**
  - **Framework:** Next.js
  - **Styling:** Tailwind CSS
- **Backend**
  - **Framework:** FastAPI
  - **Programming Language:** Python
- **Tools & Technologies**

    - **Development Environment:** Python 3.x
    - **Version Control:** Git, GitHub
    - **Deployment:** Docker for Containerization and Deployment
- **Libraries & Packages**
    - LangChain (Query Chain Creation)
    - Hugging Face Transformers (Embedding Generation)
    - Groq (LLM Integration)
    - FastAPI (API Management)
    - PyPDFLoader (Text Extraction)

6. SYSTEM DESIGN

1. Overview of the System

The project aims to streamline legal research and case file study by using Artificial Intelligence (AI) and Retrieval-Augmented Generation (RAG).
The system has three primary features:

- Summarizing case files using a RAG pipeline using LangChain, Groq, Hugging Face for embeddings, Recursive Character Text Splitter to divide the document into chunks of text and ChromaDB as a vector store.
- Allowing users to upload case files and ask queries interactively using RAG.
- Enabling users to query constitutional provisions, IPC criminal codes and what they mean and general laws and regulations using a document knowledge base.

The frontend is built using Next.js while the backend uses FastAPI to handle API requests and integrates the RAG pipeline with the application.

2. High-Level Architecture

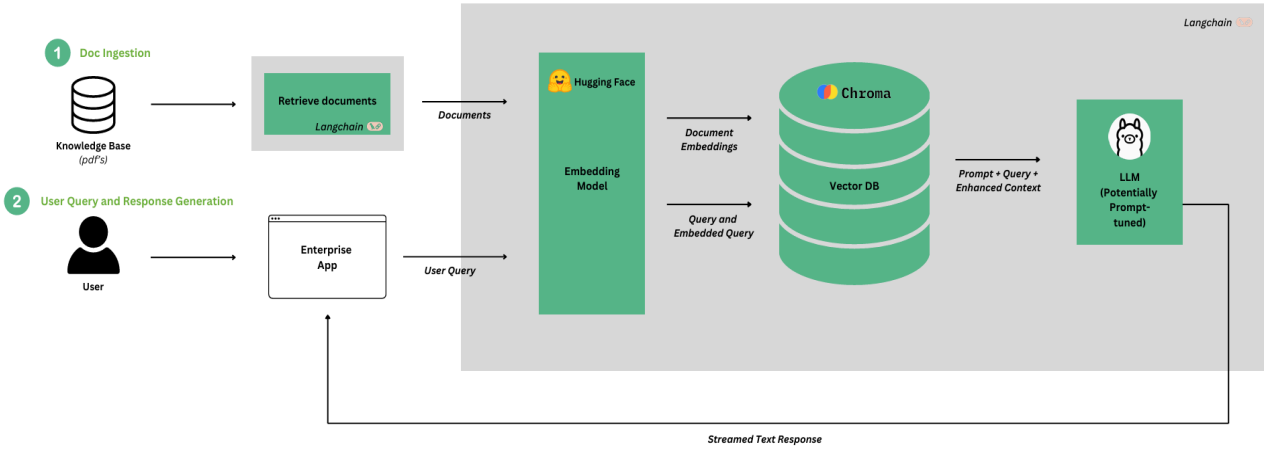

Figure 6.2: NomosAI Architecture Diagram

High-Level Architecture Components

- **Frontend (Next.js):** User interface for uploading case files, querying and receiving the response.

- **Backend (FastAPI):** API layer that manages data flow between the frontend and the RAG pipeline.
- **RAG Pipeline:** The RAG pipeline is responsible for handling querying and returning relevant responses to the user.
    - **Uploading Files:** PyPDFLoader for loading the pdf documents uploaded by the user.
    - **Embedding Generation:** Hugging Face embedding model (Sentence Transformer all-MiniLM-L6-v2).
    - **Recursive Character Text Splitter:** Splits the document into smaller chunks of text to fetch accurate results.
    - **Vector Store:** ChromaDB for storing and retrieving embeddings.
    - **Chain Creation:** LangChain used to create a chain for query formulation.
    - **Groq Integration:** Groq to using the Large Language Model (Llama 3.3) to generate a response from the data fetched by the RAG.
- **Data Storage:** Amazon DynamoDB for secure storage of case files and chat history.

3. Data Flow

Data Flow steps

1. The user uploads either a case file for summarization and querying or submit a law and regulations related query using the frontend.
2. These inputs are processed by a backend powered by FastAPI which follows a series of well-defined steps to generate accurate and meaningful responses using the RAG pipeline.
3. **For case files:**
    a. The pdf file is split into smaller chunks of text using the Recursive Character Text Splitter.
    b. Chunks are then converted into embeddings using the Hugging Face Embedding model (Sentence Transformer all-MiniLM-L6-v2).
    c. Embeddings are stored in ChromaDB for indexing.
    d. A chain is created using LangChain where the relevant chunks are fetched from the database.

e. The fetched data is fed to the Large Language Model (LLM) to generate an intelligible response.

4. **For queries:**
    a. Relevant embeddings are retrieved from ChromaDB.
    b. The query is processed and the response is generated.
5. The response is displayed to the user via frontend.

The uploaded case files and chat history are securely stored in the Amazon DynamoDB. The embeddings metadata is managed by the ChromaDB which generates a schema file with the sqlite3 extension.

4. Database Schema

The database schema generated by the ChromaDB is in the form of an sqlite file. This file contains the schema for collections, embeddings, segments, embeddings metadata, segment metadata etc.

The database schema represents a multi-tenant vector embedding management system designed to support flexible and scalable data storage with complex metadata capabilities. This schema serves as a comprehensive blueprint for organizing and managing vector embeddings across multiple contexts.

Architectural Design Principles

- **Multi-Tenancy Support:** Hierarchical structure that enables isolated data management.
- **Flexible Metadata:** Dynamic key-value metadata storage.
- **Vector Embedding Optimization:** Efficient storage and retrieval mechanisms.
- **Referential Integrity:** Strong relational constraints.

Key Structural Components

1. **Organizational Hierarchy**

    The schema implements a nested organizational structure:

    - **Tenants:** Top-level organizational units.
    - **Databases:** Associated with specific tenants.

- **Collections:** Defined within a specific database

2. **Core Data Management Table**
    a. **collections:** Stores vector collection metadata.
    b. **embeddings:** Manages individual embedding representations.
    c. **segments:** Defines granular data segmentation.
    d. **embedding_metadata:** Provides flexible metadata for embeddings.
    e. **segment_metadata:** Enables extensible segment annotations.

Architectural Design Principles

The schema utilizes foreign key constraints to establish robust data relationships:

- '*collections.database_id*' references '*databases.id*'
- '*segments.collection*' references '*collection.id*'
- '*databases.tenant_id*' references '*tenants.id*'

Unique Constraint Strategies

These constraints are implemented to ensure data integrity is maintained.

- Prevent duplicate collection names within databases.
- Ensure unique segment and embedding combinations.
- Maintain unique tenant database names.

Metadata Flexibility

Metadata tables support multiple value types:

- String values.
- Integer values.
- Float values.
- Boolean representations.

This database schema provides a robust and flexible framework for managing vector embeddings across complex and multi-tenant environments, balancing performance, scalability and metadata quality.

5. RAG Pipeline Design

The Retrieval-Augmented Generation (RAG) pipeline is a central component of the system, designed to enhance the capabilities of legal research and case file analysis by integrating retrieval mechanisms with generative models. The RAG architecture

effectively combines the strengths of information retrieval and natural language generation to provide users with accurate, contextually relevant responses to their queries.

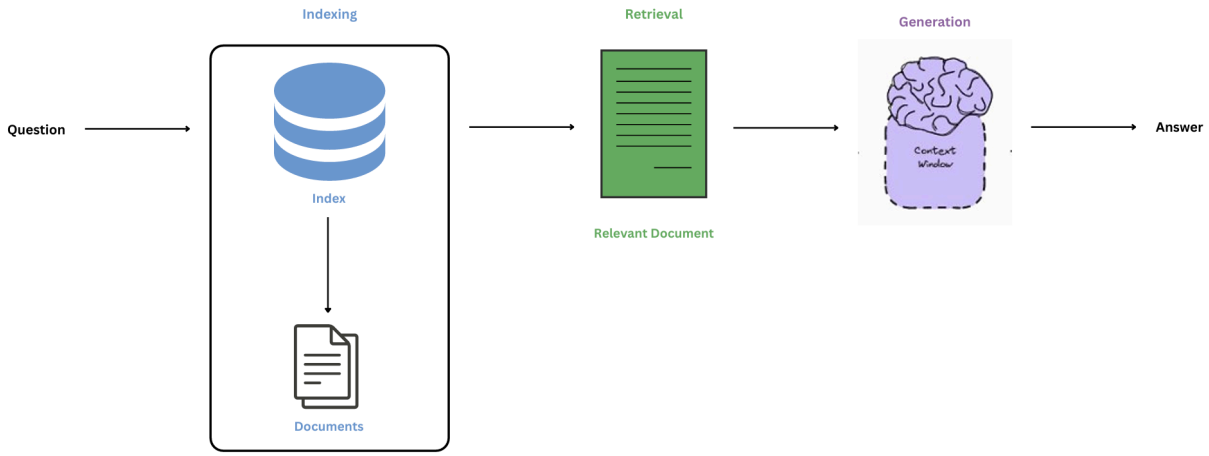

Figure 6.5.1: Basic RAG Architecture

1. **Document Ingestion**
    a. **File Upload:** Users can upload case files in PDF format through the Next.js frontend.
    b. **PDF Processing:** The PyPDFLoader is employed to extract text from uploaded PDF documents, ensuring that the content is ready for further processing.
2. **Text Splitting**
    a. **Recursive Character Text Splitter:** This component divides the extracted text into smaller, manageable chunks. By breaking down the document into segments, the system can improve retrieval accuracy and relevance when responding to user queries.
3. **Embedding Generation**
    a. **Hugging Face Embedding Model:** The Sentence Transformer all-MiniLM-L6-v2 model is utilized to convert text chunks into embeddings. These embeddings capture semantic meaning, allowing for effective similarity searches within the vector store.

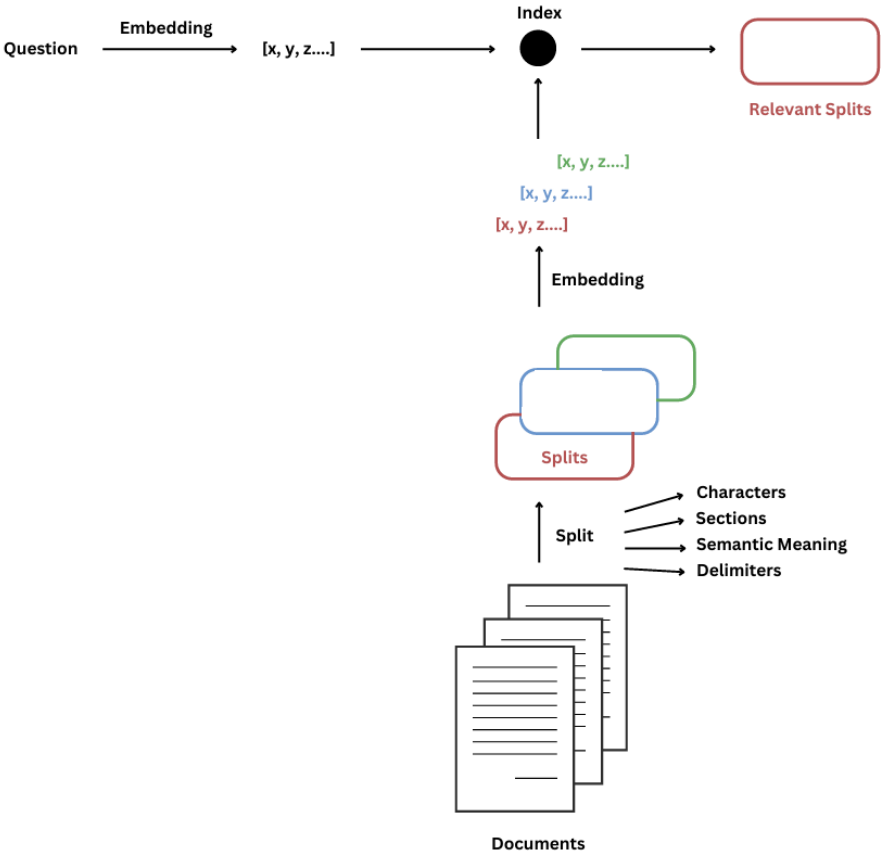

Figure 6.5.2: Ingestion, Splitting and Embedding

4. **Vector Storage**
    a. **ChromaDB:** The generated embeddings are stored in ChromaDB, which serves as a vector database. This storage solution enables efficient indexing and retrieval of embeddings based on user queries.
5. **Query Processing and Retrieval**
    a. **LangChain Integration:** LangChain is used to create a query formulation chain that interacts with ChromaDB. When a query is submitted, relevant embeddings are retrieved based on semantic similarity to the input query.
    b. **Dynamic Retrieval:** The system dynamically fetches relevant text chunks from ChromaDB that correspond to the user's query context.

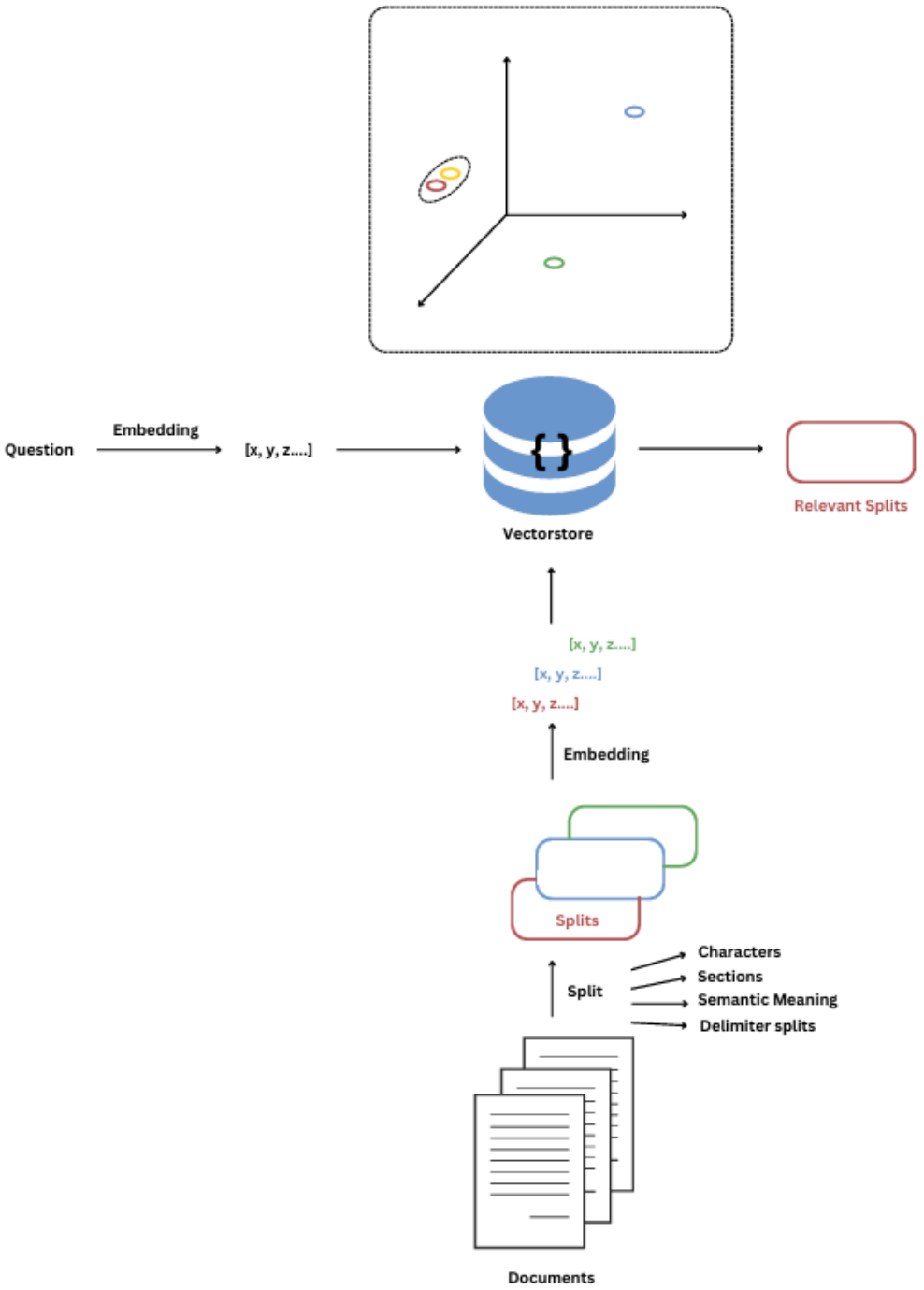

Figure 6.5.3: Query Processing and Response Retrieval

6. **Response Generation**

   a. **Groq Integration:** The retrieved data is fed into Groq, which utilizes the Large Language Model (Llama 3.3) to generate coherent and contextually appropriate responses. This integration allows for sophisticated language generation that aligns with legal terminology and concepts.

The RAG pipeline design effectively integrates various AI components to streamline legal research processes, providing users with powerful tools for summarizing case files and querying legal documents. By leveraging advanced embedding techniques and generative models, this pipeline enhances the overall user experience while maintaining high standards of accuracy and relevance in legal information retrieval.

7. PLAN OF WORK

Phase 1: Research & Literature Review

- Gather foundational materials on legal AI, RAG pipelines, and related works.
- Deep dive into legal datasets and domain-specific challenges.

Phase 2: System Analysis & Design

- Analyze existing systems to identify requirements and document challenges in legal AI applications.
- Develop high-level system architecture and design the core components (Frontend, Backend, RAG pipeline, etc.).

Phase 3: Implementation & Development

- Set up the development environment and integrate the frontend (e.g., using Next.js) with the backend (FastAPI).
- Implement the core RAG pipeline including text extraction, chunking, and embedding generation.
- Integrate the response generation component using an LLM to test retrieval and summarization functionalities.

  Deliverables:

  - A working prototype featuring document upload and a basic query interface.
  - Initial integration with a sample dataset (e.g., ChromaDB integration and embedding creation).
  - A functional system demo that processes legal documents and generates accurate summaries/answers.

Phase 4: Testing, Evaluation & Optimization

- Conduct comprehensive testing using various legal documents and queries to evaluate system performance.
- Optimize system performance by addressing issues such as hallucinations and contextual mismatches.

Deliverables:

- Detailed test cases and performance evaluation reports (including metrics like precision, recall, and ROUGE scores).

8. REFERENCES

[1] Hou, A. B., Weller, O., Qin, G., Yang, E., Lawrie, D., Holzenberger, N., Blair-Stanek, A., & Benjamin, V. D. (2024, June 24). CLERC: a dataset for Legal Case Retrieval and Retrieval-Augmented Analysis Generation. arXiv.org. https://arxiv.org/abs/2406.17186

[2] Pipitone, N., & Alami, G. H. (2024, August 19). LegalBench-RAG: a benchmark for Retrieval-Augmented Generation in the legal domain. arXiv.org. https://arxiv.org/abs/2408.10343

[3] Peng, X., & Chen, L. (2024, October 15). Athena: Retrieval-augmented Legal Judgment Prediction with Large Language Models. arXiv.org. https://arxiv.org/abs/2410.11195

[4] Louis, A., Gijs, V. D., & Spanakis, G. (2023, September 29). Interpretable Long-Form Legal Question Answering with Retrieval-Augmented Large Language Models. arXiv.org. https://arxiv.org/abs/2309.17050

[5] Wiratunga, N., Abeyratne, R., Jayawardena, L., Martin, K., Massie, S., Nkisi-Orji, I., Weerasinghe, R., Liret, A., & Fleisch, B. (2024, April 4). CBR-RAG: Case-Based Reasoning for Retrieval Augmented Generation in LLMs for Legal Question answering. arXiv.org. https://arxiv.org/abs/2404.04302

[6] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017b, June 12). *Attention is all you need*. arXiv.org. https://arxiv.org/abs/1706.03762

[7] Padiu, B., Iacob, R., Rebedea, T., & Dascalu, M. (2024). To what extent have LLMs reshaped the legal domain so far? A scoping literature review. *Information*, *15*(11), 662. https://doi.org/10.3390/info15110662