Finance Laws Retrieval-Augmented Generation (RAG) System

Ironhack Data Science and Machine Learning Bootcamp
Date: February 5, 2025
Submission Date:mFebruary 7, 2025
Authors: Ginosca Alejandro Dávila & Natanael Santiago Morales

---

Introduction

Retrieval-Augmented Generation (RAG) is an AI framework that enhances information retrieval by combining:
- A vector database to retrieve relevant documents.
- A Large Language Model (LLM) to generate contextualized responses.

This approach is widely used in legal research, finance, customer support, and knowledge management improving accuracy and relevance in automated systems.

Project Overview

This project aims to build a RAG system focused on finance laws and regulations. By integrating document retrieval and LLM-based generation, the system will provide accurate, contextualized answers to legal and financial queries.

We will experiment with different embedding models (e.g., OpenAI's text-embedding-3-large vs. all-MiniLM-L6-v2) and vector databases (e.g., ChromaDB) to optimize retrieval and response generation.

Dataset Description

- Source: A collection of finance laws, regulations, and legal texts.
- Format: .zip file containing multiple .pdf files, each representing legal provisions, case law, or financial regulations.

Goals

- Develop a RAG pipeline specialized in finance law and legal text retrieval.

- Compare OpenAI embeddings and Sentence Transformers for legal text retrieval.
- Evaluate the system based on retrieval accuracy and response relevance.

---

1. Exploratory Data Analysis (EDA)

The **Exploratory Data Analysis (EDA)** was conducted to gain insights into the dataset, identify potential preprocessing challenges, and prepare for embedding and retrieval processes in the **Retrieval-Augmented Generation (RAG) system**.

Dataset Overview

- **Dataset Type:** Collection of finance laws and regulations in **PDF format**.
- **Source:** European finance directives and legal provisions.
- **Storage Location:** Local directories.
- **Contents:** 11 PDF documents covering legal frameworks such as:
    - Anti-Money Laundering Directive (AMLD)
    - Capital Requirements Regulation (CRR)
    - Payment Services Directive (PSD)
    - Investment Firm Regulation (IFR)
    - Deposit Guarantee Scheme Directive (DGSD)
    - Securitization Regulation (SecReg)
    - Other financial and regulatory laws

EDA Process

1. File Inspection and Structure

- The dataset was extracted and inspected for file integrity.
- The extracted dataset contained a structured folder with finance law PDFs.
- A `__MACOSX` folder was detected and removed as it contained unnecessary metadata.

2. Metadata Extraction & Page Analysis

Each document was analyzed for:

- Total page count (ranging from 18 to 337 pages).
- Document titles, creation dates, and authorship metadata.
- Word count analysis revealed significant variations:
    - Largest document: *CRR_EURLEX.pdf* (220,312 words).

    - Smallest document: *WTR_EURLEX.pdf* (10,031 words).

3. Content Preview and Extraction Challenges

- Structured legal format:
    - Titles and preambles at the beginning.
    - Articles and legal clauses in the middle.
    - Annexes and correlation tables at the end.
- Text extraction challenges identified:
    - Headers, footnotes, and page numbers needed removal.
    - Presence of correlation tables requiring special handling.
    - Some documents contained long lists of legal references that needed post-processing.

Key Findings from EDA

- The dataset is well-structured but requires cleaning and preprocessing before embedding.
- Metadata extraction helped in understanding document origins, publication dates, and scope.
- Legal references and footnotes needed standardization to improve retrieval accuracy.
- Document lengths and word counts varied, impacting chunking strategies for embeddings.

---

2. Chunking Process

To ensure efficient retrieval and accuracy in response generation, we implemented a **chunking strategy** for processing the legal text.

Why Chunking is Important

- Legal texts are long and contain structured information.
- Embedding models have **token limitations** (e.g., OpenAI models have a maximum context window).
- Breaking text into **smaller, meaningful segments** ensures better retrieval results.

Chunking Strategy

We experimented with two methods:

1. `CharacterTextSplitter` → Splits text into **fixed-length chunks** with controlled overlap.
2. `RecursiveCharacterTextSplitter` → Dynamically splits text but **produces excessively small chunks**, reducing context.

Final Chunking Parameters

- **Method Chosen:** `CharacterTextSplitter`
- **Chunk Size:** 1000 characters
- **Chunk Overlap:** 200 characters

Implementation Process

1. **Extract text from each PDF document**.
2. **Split the extracted text into pages**.
3. **Apply chunking with the selected parameters**.
4. **Store chunked text in a structured format for embedding**.

---

3. Connection Setups: Embedding & Retrieval System

The RAG system integrates **LLM-based retrieval** with a **vector database** to enhance legal query response accuracy. Below are the connection setups:

Embedding Models Tested

Two main embedding models were evaluated for text vectorization:

1. `text-embedding-3-large` **(OpenAI)**
    - Performs well on complex legal text.
    - Captures semantic meaning rather than just keyword matching.
2. `all-MiniLM-L6-v2` **(Sentence Transformers)**
    - More lightweight and cost-effective.
    - Optimized for shorter legal queries.

Vector Databases Used

The system tested two main vector database options:

1. **ChromaDB** (Final choice)
    - Fast retrieval speed and scalability.
    - Easily integrates with LangChain for query processing.
2. **FAISS** (Tested but not used in final system)
    - Efficient similarity search but required additional indexing optimizations.

RAG Pipeline Overview

1. **Data Ingestion:**
    - PDF documents were extracted, cleaned, and stored.
2. **Chunking & Embedding:**
    - Legal text was split into meaningful chunks to optimize retrieval.
    - Each chunk was converted into vector embeddings.
3. **Retrieval & Query Processing:**
    - Queries were transformed into embeddings.
    - The system retrieved the most relevant legal sections using semantic search.
4. **LLM Response Generation:**
    - Retrieved text chunks were passed to a Large Language Model (LLM) for contextualized answers.

---

4. Evaluation Metrics for RAG System Performance

To assess the quality of our **Retrieval-Augmented Generation (RAG) system**, we employ a structured evaluation framework with six key metrics. Each metric is scored on a **0-1 scale** and contributes to a weighted composite score, ensuring a **quantitative and qualitative assessment** of retrieval accuracy and response relevance.

1. Correctness (40%)

Definition: Measures whether the generated response accurately reflects the content from the retrieved source documents.

- A score of **1.0** indicates a fully accurate response with no misinterpretations.
- A score of **0.0** reflects a response containing incorrect or misleading information.

Example Application: If a query asks about **financial institutions' record-keeping obligations**, the response must correctly state that **authorities have the power to**

require institutions to maintain detailed records of financial contracts. Any deviation or misrepresentation reduces the correctness score.

2. Completeness (30%)

Definition: Evaluates whether the response covers all necessary aspects of the query without omitting critical information.

- A **1.0 score** means the response fully answers the question, addressing all essential points.
- A **0.5 score** reflects partial coverage, missing minor but relevant details.
- A **0.0 score** is given if the response is largely incomplete.

Example Application: If a query asks about **credit risk mitigation under different financial models**, the response must outline how mitigation is applied under both the **Standardized Approach** and the **Internal Ratings-Based (IRB) Approach**, specifying relevant articles and calculations.

3. Conciseness (10%)

Definition: Ensures that the response is succinct and free of unnecessary elaboration or unrelated details.

- A **1.0 score** is awarded for precise answers without redundancy.
- A **0.5 score** is given for slightly verbose responses that contain some extraneous information.
- A **0.0 score** is assigned to excessively long or unfocused responses.

Example Application: A query about **RtC K-factor requirements** should yield a response stating the formula and component definitions **without excessive background explanation** beyond the requested details.

4. Relevance (10%)

Definition: Assesses whether the response strictly pertains to the retrieved legal documents and the user's query.

- A **1.0 score** indicates the response remains fully on-topic.

- A **0.5 score** is given if minor irrelevant details are included.
- A **0.0 score** is assigned when the response contains substantial off-topic information.

Example Application: If the user asks about **DGS investment regulations**, the response should explicitly state that **Deposit Guarantee Schemes should invest in low-risk assets**. Irrelevant mentions of financial institution obligations would lower the relevance score.

5. Citation Accuracy (5%)

Definition: Measures whether the response correctly cites the **exact source document and page number** for retrieved information.

- A **1.0 score** is awarded when citations precisely match the referenced content.
- A **0.5 score** is given if citations are slightly off (e.g., incorrect page but correct document).
- A **0.0 score** is assigned if citations are missing or entirely incorrect.

Example Application: If a query references a specific regulation on **financial transparency (Article 108)**, the response must **cite the correct page and document name** from the official source.

6. Language Quality (5%)

Definition: Evaluates the clarity, grammar, and readability of the generated response.

- A **1.0 score** is given for grammatically sound, professional, and well-structured responses.
- A **0.5 score** is assigned for responses with minor grammatical errors or awkward phrasing.
- A **0.0 score** is given for responses with significant grammatical issues or poor readability.

Example Application: Responses should be structured in **clear, professional legal language**, free from spelling or syntax errors that could affect comprehension.

Final Weighted Score Calculation

To derive an overall **performance score**, the individual scores are multiplied by their respective weights:

$$Final\ Score\ = (0.4 \times Correctness) + (0.3 \times Completeness) + (0.1 \times Conciseness) + (0.1 \times Relevance) + (0.05 \times Citation\ Accuracy) + (0.05 \times Language\ Quality)$$

This weighted evaluation ensures a **balanced and rigorous assessment** of our RAG system's legal research capabilities, identifying areas for **optimization and improvement**.

---

5. System Performance Evaluation

To assess the performance of Hugging Face and OpenAI models, we evaluated responses for four questions (Q1 to Q4) using six key metrics: **Correctness, Completeness, Conciseness, Relevance, Citation Accuracy, and Language Quality**. Each metric was assigned a weight based on its importance in determining overall response quality. The weighted scores were then computed for each question and model.

The following table summarizes the performance results:

Metric	Weight	Q1 Hugging Face	Q1 OpenAI	Q2 Hugging Face	Q2 OpenAI	Q3 Hugging Face	Q3 OpenAI	Q4 Hugging Face	Q4 OpenAI
Correctness	0.4	0.9	0.8	0.9	0.9	0.85	0.9	0.95	0.75
Completeness	0.3	0.95	0.85	0.85	0.95	0.9	0.85	0.9	0.7
Conciseness	0.1	0.9	0.7	0.9	0.75	0.8	0.75	0.9	0.75
Relevance	0.1	0.9	0.75	0.9	0.85	0.9	0.85	0.9	0.75
Citation Accuracy	0.05	0.2	0.1	0.9	0.95	0.6	0.85	0.95	0.6
Language Quality	0.05	0.95	0.9	0.95	0.9	0.95	0.9	0.9	0.9
Weighted Score	-	0.88	0.78	0.8875	0.8975	0.865	0.86	0.9225	0.735

Key Findings and Insights

- Hugging Face consistently outperforms OpenAI in retrieval accuracy and citation precision across all four legal queries.
- OpenAI embeddings tend to generalize more, leading to lower retrieval accuracy and citation precision. While its responses are more fluent, they sometimes introduce unrelated legal details that reduce relevance.
- Weighted scores confirm that Hugging Face is the preferred option for legal queries requiring high precision and correct citations.
- OpenAI performed best in Q2, where it provided a more detailed response. However, Hugging Face still delivered more accurate references to the source material.

Conclusion

For general finance-related Q&A, OpenAI may be useful due to its faster response time. However, for high-stakes legal document retrieval, Hugging Face embeddings provide superior accuracy and citation reliability.

---

6. Challenges & Recommendations

Challenges Encountered

- Text extraction inconsistencies (e.g., headers/footers remained in some cases).
- Handling of correlation tables was difficult during text parsing.
- Document chunking strategy impacted retrieval quality.

Recommendations for Future Improvements

- Fine-tuned embedding models specifically trained on legal documents could further improve retrieval.
- Hybrid chunking approaches (fixed + semantic-based) may provide better retrieval granularity.
- Expanding the dataset to include legal case studies & financial reports can enrich context understanding.

- Deployment as a web app (Flask/Streamlit) would make the system more accessible.

Final Thoughts

The **Finance Laws RAG System** successfully integrates retrieval-augmented generation with vector search to enhance legal research capabilities. By leveraging LLMs and vector databases, the system efficiently retrieves and summarizes complex financial regulations, offering a powerful legal information retrieval tool.

Future iterations will focus on fine-tuning the embeddings, enhancing response structuring, and expanding the dataset for broader financial regulatory coverage.