Retreival Augmented Generation from Scratch for Large Language Models

Sarim Aeyzaz - (i21-0328)

Abstract

This project aimed to develop a scratch implementation of retrieval augmented generation system using language models and a custom vector database integration. The primary objective was to enhance question-answering capabilities by leveraging pre-trained language embedding and an efficient document storage mechanism.

The methodology involved utilizing the SentenceTransformer library to embed documents into dense vectors and integrating these embeddings with the Chroma database for efficient similarity search. PDF documents were processed to extract text, which was then segmented into sentence chunks for indexing.

Key findings include the successful implementation of a Retrieval-Augmented Generation (RAG) model that can respond to user queries based on relevant document chunks stored in the Chroma database. The system visually has shown to tell answers that align more with the textbook contents.

In conclusion, the project highlights the potential of combining state-of-the-art language models with custom database solutions for advanced information retrieval tasks. The developed system offers a scalable and efficient approach to document retrieval and question answering, with implications for various applications in natural language processing and information retrieval.

1 Introduction

The rapid growth of digital text data, particularly in the form of unstructured documents such as PDFs, presents significant challenges for effective information retrieval and question-answering systems. Traditional keyword-based approaches often fall short in capturing the semantic meaning and context of natural language queries. This project aims to address these challenges by leveraging advanced language models and custom database integration to enhance document retrieval and question-answering capabilities.

In the field of natural language processing (NLP), the ability to efficiently retrieve and analyze information from large text corpora is essential for various applications, including information retrieval, summarization, and question-answering. However, existing methods often struggle with understanding the nuanced relationships and context within text data.

The problem statement revolves around improving the accuracy and relevance of information retrieval systems, particularly when dealing with complex queries and large document collections. By harnessing the power of state-of-the-art language embeddings and integrating them with an optimized database structure, we aim to develop a Retrieval-Augmented Generation (RAG) model that can effectively retrieve and generate responses based on the context and semantics of user queries.

The significance of this work lies in its potential to revolutionize how we interact with textual data, enabling more intuitive and accurate information retrieval systems that better understand the nuances of natural language. This project contributes to advancing the field of NLP by demonstrating a practical approach to combining cutting-edge language models with efficient database technologies for enhanced document retrieval and question-answering.

2 Methodology

The project leverages a combination of advanced NLP tools and custom integration to achieve efficient document retrieval and question-answering capabilities. The methodology is outlined as follows:

2.1 Data Collection and Preprocessing

The initial phase involves collecting and preprocessing textual data from PDF documents. This is achieved using the Fitz library for PDF parsing and the Spacy library for text processing tasks such as sentence tokenization and chunking. The majority of this implementation is done from scratch.

2.2 Sentence Embedding with SentenceTransformer

To convert textual data into numerical embeddings suitable for similarity search, we utilize the SentenceTransformer library. This library provides pre-trained models for generating fixed-size embeddings from sentences. We employ the all-MiniLM-L12-v2 model due to its efficiency and competitive performance. You can check it's position in the leaderboard at: https://www.sbert.net/docs/pretrained_models.html

2.3 Chroma Database Integration

The Chroma database serves as a persistent storage solution for the preprocessed text data and associated embeddings. Integration with the Chroma database is achieved using a custom wrapper class (SentenceTransformerEmbeddings) that adapts the SentenceTransformer model for compatibility with Chroma's embedding function style. This integration enables efficient storage, retrieval, and similarity search operations on the text data.

2.4 Retrieval-Augmented Generation (RAG) Model

The core component of the project is the RAG model, which combines document retrieval with language generation techniques. The RAG model is implemented using the Ollama library, specifically the phi3 model https://ollama.com/library/phi3, which is already optimized for question-answering tasks. The model retrieves relevant document contexts using the Chroma database and generates responses based on user queries and retrieved contexts.

2.5 Implementation Details

The project is implemented in Python, utilizing popular libraries such as spaCy, fitz, SentenceTransformer, Chroma, and Ollama. The workflow involves sequential data processing steps, starting from PDF parsing and preprocessing, followed by sentence embedding, database integration, and finally, the implementation of the RAG model for question answering.

3 Data Collection and Preprocessing

The data collection and preprocessing phase focuses on extracting textual information from PDF documents and preparing it for further analysis and NLP tasks. The following steps outline the process:

3.1 PDF Parsing

PDF parsing is performed using the fitz library, which allows for efficient extraction of text content from PDF files. Each PDF document is opened using fitz.open() to facilitate page-wise processing.

3.2 Text Extraction

For each page of the PDF document, the text content is extracted using the get-text() method provided by the fitz library. This raw text is then cleaned by removing unnecessary characters such as newline characters and leading/trailing spaces.

3.3 Tokenization and Sentence Chunking

The extracted text undergoes tokenization and sentence chunking using the spaCy library. Here are the steps involved:

Sentence Tokenization: The spaCy English model (English) is used to tokenize the extracted text into individual sentences. This is achieved by creating a spaCy pipeline with a sentence tokenizer (sentencizer) added as a component.

Text Segmentation: Each page's text is segmented into sentences, where each sentence is represented as a string. This segmentation allows for finer-grained analysis of textual data at the sentence level.

3.4 Data Representation

The processed text data is represented in a structured format, typically as a list of dictionaries or objects, where each entry corresponds to a page in the PDF document. Each entry contains metadata such as the source PDF file, page number, and a list of segmented sentences.

3.5 Data Cleaning and Normalization

During preprocessing, additional cleaning and normalization steps may be applied to improve the quality of the extracted text. This can include removing special characters, handling hyphenated words, and correcting common formatting issues.

4 Results and Analysis

The RAG (Retrieval-Augmented Generation) model was evaluated for its performance in responding to queries using the integrated Chroma database. The experiments involved querying the RAG model with various questions related to the extracted text from PDF documents and analyzing the generated responses.

4.1 Querying RAG Model

To assess the RAG model's performance, several queries were posed to the system. These queries covered a range of topics related to the content extracted from the PDF documents, including technical concepts, definitions, and comparisons. Examples of queries used in the evaluation include:

"What is the difference between TCP and UDP?" "Explain the concept of natural language processing." "How does machine learning differ from deep learning?"

4.2 Response Evaluation

The responses generated by the RAG model were evaluated based on their relevance, coherence, and accuracy. Key metrics considered during the evaluation included:

Relevance: Assessing whether the response addresses the query's intent and provides meaningful information related to the topic.

Coherence: Examining the logical flow and structure of the response to ensure it is well-organized and easy to understand.

Accuracy: Verifying the factual correctness of the information provided in the response, particularly for technical or domain-specific queries.

4.3 Chroma Database Integration

The integration of the Chroma database with the RAG model facilitated efficient information retrieval based on query similarity. The Chroma database leverages semantic embeddings of text chunks extracted from PDF documents, enabling rapid retrieval of relevant content.

4.4 Performance Metrics

Performance metrics such as retrieval accuracy, response quality, and user satisfaction were used to evaluate the overall effectiveness of the RAG model in conjunction with the Chroma database.

4.5 Analysis of Findings

The analysis of findings highlighted the strengths and limitations of the RAG model in responding to queries based on the extracted PDF content. Insights gained from the evaluation process provided valuable feedback for optimizing the model's performance and refining the integration with the Chroma database.

5 Discussion

The project utilized a combination of the Sentence-Transformer library, RAG (Retrieval-Augmented Generation) model, and Chroma database integration to facilitate information retrieval and question answering based on PDF document content. This section delves into the strengths and limitations of the approach, along with insights gained during implementation.

5.1 Strengths

The key strengths of the project include:

Efficient Information Retrieval: The integration of the Chroma database with the RAG model enabled rapid retrieval of relevant information from PDF documents based on query similarity.

Semantic Embeddings: Leveraging Sentence-Transformer for generating semantic embeddings allowed for accurate representation of text chunks, enhancing the quality of retrieval and response generation.

Scalability: The modular design of the system allows for scalability, enabling the addition of more PDF documents and expansion of the knowledge base without significant performance overhead.

Flexible Querying: The RAG model supports flexible querying, allowing users to pose natural language questions and receive contextually relevant responses.

5.2 Limitations

Despite its strengths, the project has certain limitations:

Token Length Constraints: The SentenceTransformer library imposes token length constraints, which may limit the effectiveness of sentence chunking and retrieval for longer text segments.

PDF Extraction Challenges: Extracting text from PDF documents can be challenging, especially with complex formatting, images, or scanned documents, which may affect the quality of extracted content.

Semantic Understanding: While semantic embeddings enhance information retrieval, nuances in natural language understanding and contextual relevance remain challenging aspects.

5.3 Challenges Encountered

During implementation, several challenges were encountered:

PDF Parsing Complexity: Parsing and extracting text from PDF documents with diverse layouts and structures required robust preprocessing techniques to ensure accurate content extraction.

Integration Complexity: Integrating the RAG model with the Chroma database and ensuring seamless interoperability involved addressing compatibility issues and fine-tuning system configurations.

5.4 Future Improvements

To enhance the project's effectiveness and address its limitations, the following areas for future improvement can be considered:

Enhanced PDF Processing: Implement advanced PDF processing techniques to handle complex document structures, including image-to-text conversion and OCR (Optical Character Recognition) for scanned documents. Since the DIP book, it struggled to convert formulas to a proper representation.

Model Optimization: Maybe work on somehow optimizing the sentence transformer's parameters(?).

User Interface Enhancement: Develop a user-friendly interface to streamline query input, visualize retrieved content, and facilitate interactive exploration of the PDF document knowledge base.

Evaluation and Feedback: Conduct comprehensive user evaluations and gather feedback to iteratively refine the system based on real-world usage scenarios and user requirements.

6 Conclusion

In conclusion, this project successfully demonstrated the integration of advanced natural language processing techniques to enable efficient information retrieval and question-answering from PDF document collections. The key findings and contributions of this work include:

Implementation of the RAG (Retrieval-Augmented Generation) model facilitated by the SentenceTransformer library for semantic text embeddings.

Integration of the Chroma database for storing and retrieving structured text data based on semantic similarity.

Evaluation of the system's performance in responding to natural language queries, showcasing its potential for real-world applications in knowledge management and information retrieval tasks.

The project underscores the significance of leveraging state-of-the-art NLP tools and techniques for unlocking insights from unstructured textual data, with implications for diverse domains such as academia, research, and industry.

7 References

1. **Phi3:**
   - URL: https://huggingface.co/docs/transformers/main/en/m
2. **RAG Paper:**
   - URL: https://arxiv.org/abs/2005.11401
3. **All-MiniLM-L12-v2:**
   - URL: https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2