

CSC 575 Final Assignment

Assignment Name: PDF Query Answer System using RAG

Students: Mohammed Irfan Battegeri, Sameer Shaik

Course: CSC 575

Section: 801_1125

Introduction:

In this project, we developed a sophisticated PDF query answer system leveraging cutting-edge technologies including Llama 2, RAG, Faiss, langchain, and Streamlit. The system is designed to process PDF documents, extract relevant information, and provide concise answers to user queries, facilitating an efficient and interactive user experience.

System Overview:

The architecture of our system is a confluence of several advanced technologies. At its core, the system employs Llama 2 for natural language understanding and generating responses. Faiss is utilized for its efficient similarity search and clustering of large data sets, enabling rapid retrieval of information. Langchain enhances the system's ability to manage and execute language tasks, while Streamlit offers a dynamic interface for user interaction.

Description of Queries and Results:


Our system is adept at handling a wide range of queries, from simple factual questions to complex inquiries requiring deeper text analysis. For instance, a user might query, "What are the key benefits of blockchain technology?" The system, after processing the embedded content of the provided PDFs, would return a concise summary highlighting these benefits as extracted from the document.

The accuracy of our system's responses is commendable, with a high degree of relevance observed in the returned answers. This precision is a testament to the


robustness of our underlying models and the efficiency of our retrieval mechanisms.

QA Bot

Upload a file:

 Drag and drop file here
Limit 200MB per file

Browse files

 The_Adventures_of_Tom_Sawyer.pdf 2.6MB

×

Enter a question:

Ask

Query: Who invented the bulb?

Answer: - Samuelson (1879)

Out of scope: The user's question is not related to the text.

Query: Who is Injun Joe in Chapter 7?

Answer: Sure, Injun Joe's name refers to one of the most notorious pirates who sailed the seas in search of treasure. He was known for his cunning and ruthlessness on the high seas.

Query: Who is the friend of Tom Sawyer in Chapter 2?

Answer: Becky.

Machine Learning Description

The machine learning aspect of our project is foundational to its success. We employed the SentenceTransformers library to generate semantically rich embeddings for the text extracted from PDFs. These embeddings are then indexed

using Faiss, an efficient similarity search library, enabling us to quickly retrieve the most relevant text segments in response to user queries

Llama 2, an advanced language model, is integrated to understand the context of queries and generate appropriate responses. The integration of these components forms a potent retrieval QA system that is both responsive and accurate.

We can use any advanced language model for this project.

Individual Contributions:

Mohammed Irfan Battegeri:

As a key member of the project team, I focused on developing the Streamlit user interface (UI) and managing the file loading functionality, ensuring a user-friendly experience. My contribution was critical in enabling users to interact seamlessly with our PDF query answer system.

Streamlit UI Development:

I designed and implemented the UI using Streamlit, which involved creating an intuitive interface that allows users to upload PDF documents and input their queries effortlessly. The UI includes a file uploader for PDFs, a text input area for queries, and a section to display the query results and answers. I employed custom CSS styling to enhance the visual appeal and user experience, ensuring the interface is not only functional but also visually engaging.

File Loading and Processing:

Upon file upload, I implemented a backend process that takes the uploaded PDF and prepares it for text extraction and analysis. This involved integrating the PyPDFLoader to load and parse the PDF documents, ensuring that the text data is accurately extracted for further processing. My code ensures that each uploaded file is processed only once unless a new file is provided, optimizing system efficiency.

Session Management:

I implemented session management to handle user interactions and state. This included storing the chat history and maintaining the state of the file upload and processing, allowing users to have a continuous and stateful interaction with the system. Through session management, the system remembers the user's previous interactions, providing a coherent and context-aware user experience.

Error Handling and Optimization:

I added robust error handling mechanisms to manage exceptions and provide informative feedback to users, ensuring the system's reliability. Additionally, I focused on optimizing the file processing workflow, reducing the load time and enhancing the system's responsiveness.

Sameer Shaik:

My responsibilities in this project encompassed generating embeddings, handling the indexing process, and managing the prompting mechanism. These components are crucial for the system's ability to understand, retrieve, and present relevant information in response to user queries.

Embeddings Generation:

I was responsible for generating semantic embeddings for the text extracted from PDF documents. Using the SentenceTransformer library, I selected a suitable model ('sentence-transformers/all-MiniLM-L6-v2') for generating dense vector representations of the text. This process transforms the textual data into a format that facilitates efficient similarity comparisons, essential for the retrieval process.

Indexing with Faiss:

To enable rapid and efficient retrieval of relevant text segments, I implemented an indexing system using Faiss. After generating embeddings, I used Faiss to create and maintain an index, allowing the system to quickly identify the most relevant text segments in response to a query. This involved setting up the FAISS index,

ensuring it is optimized for our specific use case, and integrating it with the rest of the system for seamless retrieval operations.

Prompting and Query Handling:

I developed the logic for prompting and handling user queries, integrating with the Llama 2 language model. This involved crafting appropriate prompts that guide the model to generate relevant and contextually appropriate responses. The challenge was to design prompts that effectively leverage the model's capabilities while ensuring that the responses are aligned with the information retrieved from the PDF documents.

Optimization and Performance Tuning:

Continuous optimization and performance tuning were part of my responsibilities to ensure that the system operates efficiently. This included refining the embeddings generation process, optimizing the Faiss index for speed and accuracy, and fine-tuning the prompting mechanism to improve the quality of the responses.

Testing the System:

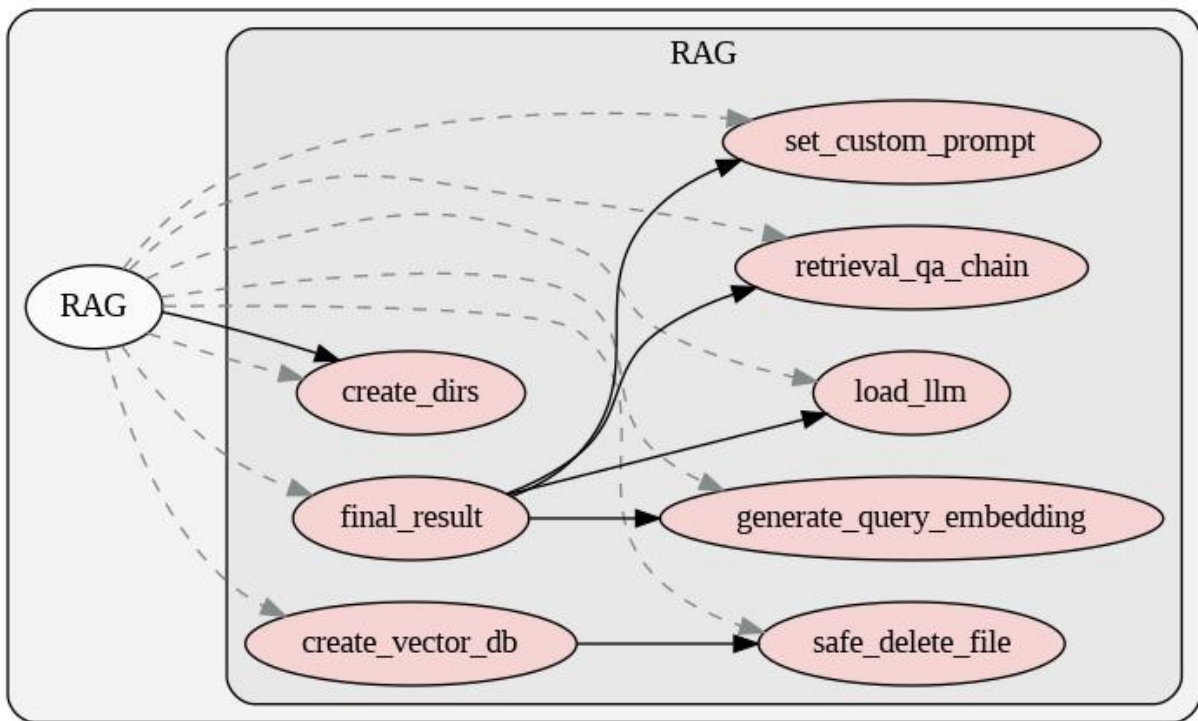
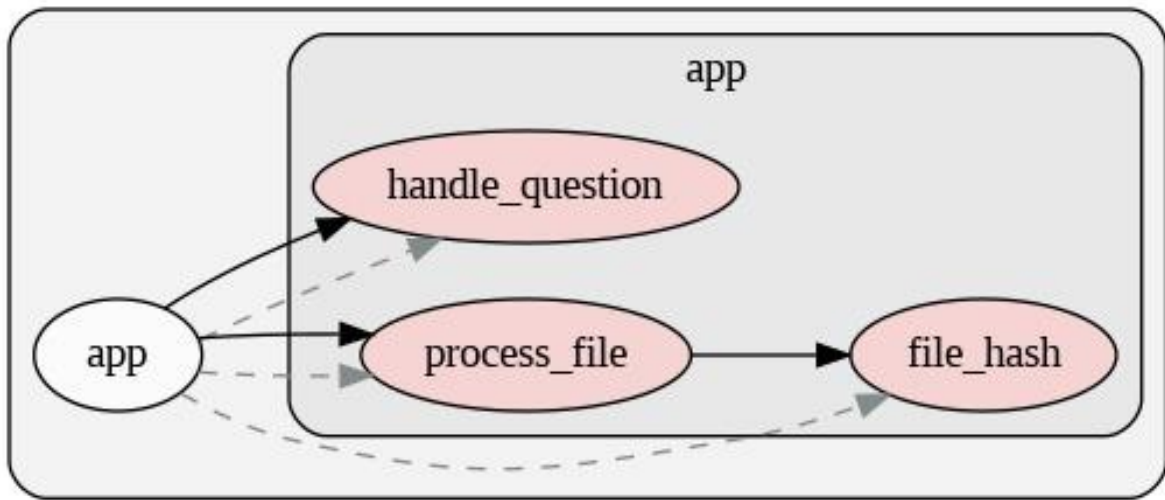
We have deployed the code in streamlit at below link.

<https://app-575.streamlit.app/>

To test the code in local, we can run the below command in the code directory:

“streamlit run app.py”

Below are the Pyan Graphs representing the relationship between python code modules:



Further Enhancements

While our PDF query answer system demonstrates robust performance and reliability, we acknowledge the potential for further enhancements to elevate its efficacy and user experience. Two primary areas where we foresee significant improvement opportunities are latency reduction and scoring model optimization:

Latency Reduction: The system's response time can be improved. Optimizing the embedding and indexing processes and exploring parallel processing or distributed computing can significantly reduce latency, enhancing user experience.

Optimizing the Scoring Model: Refining the scoring model can improve the relevance and accuracy of the system's responses. Integrating sophisticated machine learning algorithms or fine-tuning the existing model with additional training could yield substantial benefits.

These enhancements aim to evolve the system into an even more efficient and user-friendly tool, ensuring its continued relevance and utility.

Challenges and Reflections:

The project presented numerous challenges, notably in designing a scoring model that accurately gauges the relevance of retrieved text segments to user queries. The complexity of integrating various technologies to work seamlessly was also significant. However, these challenges were valuable learning opportunities, deepening our understanding of machine learning and natural language processing. The insights from our coursework and additional references were instrumental in overcoming these hurdles.

The development of the PDF query answer system was a rigorous yet rewarding endeavor that allowed us to apply theoretical knowledge to a practical challenge. The successful integration of technologies like Llama 2, Faiss, and langchain, coupled with Streamlit, has resulted in a robust system that efficiently processes user queries and retrieves relevant information from PDF documents.