Task: Implementing Retrieval Augmented Generation (RAG) using LangChain

Objective:

The objective of this task is to implement a Retrieval Augmented Generation (RAG) system using LangChain. This involves utilizing LangChain to process text data from PDFs, converting it into semantic chunks, generating embed dings, and integrating them into a vector database or memory. The system will then be able to take user queries, convert them into embeddings, retrieve relevant semantic chunks using similarity matching, and utilize them as context for a large language model (LLM) to generate responses.

Requirements:

1. Utilize LangChain for text processing, semantic chunking, and embedding generation.
2. Incorporate PDF documents as private context for the system.
3. Implement a user interface (UI) using Streamlit or similar technology to demonstrate the working demo (optional but recommended for a comprehensive presentation).

General Flow of RAG:

The general flow of RAG is generally as below. However, you are free to choose some other flow according to your convenience, as long as the requirements are completed.

1. Text Processing:
    - Convert PDF documents into text format.
    - Break down the text into semantic chunks.
2. Embedding Generation:
    - Generate embeddings for the semantic chunks.
    - Save the embeddings to a vector database or hold them in

memory. 3. User Query Processing:

    - Receive user queries.
    - Convert the queries into embeddings.
4. Similarity Matching:
    - Retrieve semantic chunks similar to the user query using similarity matching techniques.

5. Context Integration:
    - Feed the retrieved semantic chunks as context to a large language model (LLM).
6. Response Generation:
    - Utilize the context provided by the semantic chunks to generate responses from the LLM.
    - Ensure the generated prompts are well-crafted to guide the LLM and prevent hallucination or irrelevant responses.

Deliverables:

1. Code implementation of the RAG system using LangChain.
2. Documentation outlining the process flow, algorithms used, and any additional details. This doesn't need to be a comprehensive documentation, simple comments or a markdown text is enough.
3. Optional: User interface (UI) implemented using Streamlit or similar technology for demonstration purposes.

Evaluation Criteria:

1. Correct implementation of the RAG system with LangChain.
2. Efficiency and accuracy in text processing, embedding generation, and retrieval.
3. Effectiveness of the user interface (if implemented) in demonstrating the system's functionality.
4. Quality of documentation provided.
5. Creativity in prompt crafting to guide the LLM effectively.

Submission Deadline:

You have one week from the date of receiving this task to submit your implementation. However, the completion of the task is not the ultimate goal, individual implementation will be definitely considered and ac knowledged

Note: Feel free to reach out for clarification or assistance during the implementation process. Good luck!