LangChain & NLP Task

Task-2

1.Introduction

This task involves integrating an open-source Large Language Model (LLM) with LangChain to create a robust Question-Answering (QA) system. The system is designed to allow users to upload a PDF document, ask questions about its content, and receive relevant responses based on the document's context. Additionally, the system includes a general-purpose chatbot that can answer questions unrelated to any document.

2.Implementation

1.Setting Up the Models and Embeddings

- The code utilizes Google Generative AI for both primary and fallback models (gemini-1.5-flash) to process the input questions and provide answers.
- GoogleGenerativeAIEmbeddings are used to generate embeddings for the context, which aids in retrieving the most relevant text chunks for answering the question.

2.Loading and Processing PDFs

- PDF Validation: The validate_pdf function ensures that the uploaded file is a valid PDF by checking if it contains any pages.
- Text Extraction: The load_pdf function reads the PDF using pdfplumber, extracting text from each page. In cases where a page has no text, a message indicating "No text found on this page" is returned.

3.Document Splitting and Vector Indexing

- The content extracted from the PDF is divided into smaller, manageable chunks using the RecursiveCharacterTextSplitter to avoid overwhelming the language model with large input sizes.
- Each chunk is transformed into vector embeddings using the Google Generative AI embeddings, and a Chroma vector store is built to efficiently retrieve relevant chunks for question-answering.

4.Question-Answering Mechanism

- The content extracted from the PDF is divided into smaller, manageable chunks using the RecursiveCharacterTextSplitter to avoid overwhelming the language model with large input sizes.
- Each chunk is transformed into vector embeddings using the Google Generative AI embeddings, and a Chroma vector store is built to efficiently retrieve relevant chunks for question-answering.

5.Text Cleaning

- The retrieved answer is processed to clean unnecessary characters such as asterisks (*) and hashes (#), ensuring that the output is user-friendly and clear.

6.Dual Functionality: PDF Q&A and Chatbot

- The user interface gives users a choice between two options:
- PDF Q&A: Allows users to upload a PDF and ask questions specifically related to the content within the document.
- Chatbot: Provides a general-purpose chatbot to answer questions not based on the PDF.
- Depending on the user's choice, the system either loads the PDF and answers questions based on its content, or responds using the fallback chatbot model for general questions.

3.LangChain Integration

- LangChain’s Document Loaders: The code uses LangChain's text-splitting and retrieval mechanisms to process the PDF efficiently. LangChain’s Chroma vector store is responsible for creating an index and retrieving relevant sections from the text.
- Retriever Chains: The RetrievalQA chain is utilized to fetch the relevant sections from the vector index and then generate answers based on the retrieved context.

4.Conclusion

In this task,I successfully integrated an open-source Large Language Model(Google Gemini Variant) with LangChain to build a versatile Question-Answering (QA) system. The system offers dual functionality: users can upload a PDF and ask questions specifically related to its content, or use the chatbot for general-purpose queries. Through LangChain's document loaders, text splitting, vector embeddings(ChromaDB), and retriever chains, the system ensures efficient extraction of relevant information from the PDF. Additionally, the fallback chatbot model provides answers based on a broader knowledge base when needed.