Gen AI Engineer / Machine Learning Engineer Assignment

Project Overview

The "**Chat with PDF using Gemini Pro**" project provides an interactive interface that allows users to upload PDF documents and engage with a Question-Answering (QA) chatbot. This chatbot utilizes advanced AI techniques to process the documents, extract meaningful information, and respond to user queries based on the content.

Features

- **PDF Upload:** Users can upload multiple PDF files for processing.
- **PDF Processing:** Extracts and indexes text from the PDFs for efficient retrieval.
- **Embeddings:** Leverages Google Generative AI for creating semantic embeddings of the text.
- **Vector Search:** Utilizes FAISS to conduct fast searches for relevant document segments based on user queries.
- **Question Answering:** Generates answers from the retrieved document context, with fallback responses when the information is unavailable.

Project Structure

- **chatbot.py:** Contains the core logic for processing PDFs, embedding text, and generating answers.
- **app.py:** The Streamlit frontend application for user interactions, including PDF uploads and question submission.

How It Works

PDF Text Extraction

1. **Text Extraction:** Utilizes the PyPDF2 library to read and extract text from uploaded PDF files.
2. **Text Splitting:** Employs **RecursiveCharacterTextSplitter** from **Langchain** to split the extracted text into manageable chunks for efficient processing.

Embeddings and Vector Store

1. **Creating Embeddings:** Uses **GoogleGenerativeAIEmbeddings** to convert text chunks into semantic vectors.
2. **Storing Vectors:** Stores these embeddings in a **FAISS vector store**, enabling fast retrieval during the QA process.

Question Answering

1. **User Query Processing:** When a user submits a question, the system performs a similarity search in the vector store.
2. **Retrieving Context:** The retrieved document segments are passed through a QA chain using the **ChatGoogleGenerativeAI** model, generating a context-aware response.
3. **Fallback Mechanism:** If the model lacks sufficient information, it responds with "The information is not available in the document."

Usage

- **Upload PDFs:** Use the sidebar to upload your PDF files.
- **Process PDFs:** Click "Submit & Process" to extract and index the content.
- **Ask Questions:** Enter your question in the input box and submit it. The chatbot will provide an answer based on the uploaded document content.

Technologies Used

- **Python:** Primary programming language for backend logic.
- **Langchain:** Framework for document processing and creating the QA chain.
- **FAISS:** Vector search engine for efficient document retrieval.
- **Google Generative AI (Gemini Pro):** Provides embeddings and generates conversational responses.
- **Streamlit:** Frontend framework for the web interface.
- **PyPDF2:** Library for PDF text extraction.
- **dotenv:** Manages environment variables.

.

Challenges Faced

- **Text Extraction:** Some PDF formats presented challenges in extracting clean text, requiring adjustments to the extraction logic.
- **Generative Model**: Ensuring that the model provides contextually relevant responses based on retrieved information.