Adaptive Recommendation Chatbot with RAG and Vector Database

Project Report

Introduction

The InsightBot AI project was aimed at developing an adaptive recommendation chatbot using Retrieval-Augmented Generation (RAG) and a vector database. The project allows users to upload PDF and JSON files, process the text, and ask questions based on the content of the files.

Approach Taken

The approach taken for this project involved the following steps:

1. **Loading Environment Variables**: Using the `dotenv` library to load the Google API key securely.
2. **Reading Files**: Handling PDF files using `PyPDF2` and JSON files using the `json` library.
3. **Text Chunking**: Splitting the text into manageable chunks using `langchain`'s `RecursiveCharacterTextSplitter`.
4. **Embedding Text**: Using Google Generative AI Embeddings to embed the text chunks into vector representations.
5. **Creating Vector Store**: Storing the embeddings in a FAISS vector database for efficient similarity search.
6. **Conversational Chain**: Setting up a conversational chain using `langchain` to handle question-answering based on the embedded text.
7. **Streamlit Interface**: Building an interactive user interface with Streamlit for file uploads and question input.

Challenges Faced

The following challenges were encountered during the project:

1. **Rate Limiting**: The Google Generative AI API has strict rate limits, which caused quota exceeded errors during the embedding process.
2. **Handling Large Files**: Processing large files efficiently without exceeding API rate limits or running into memory issues.

3. **Ensuring Accuracy**: Maintaining the accuracy of the question-answering system while handling diverse file formats and content types.

Overcoming Challenges

The challenges were overcome using the following strategies:

1. **Batch Processing with Delays**: To handle rate limiting, text chunks were processed in smaller batches with delays between each batch.
2. **Retry Logic**: Implemented retry logic to handle rate limit errors by waiting and retrying the requests.
3. **Monitoring and Adjustments**: Regularly monitored the processing time and adjusted batch sizes and delays to optimize performance.
4. **Enhanced Error Handling**: Improved error handling to gracefully manage different types of errors and provide meaningful feedback to users.

Conclusion

The InsightBot AI project successfully developed a recommendation chatbot using RAG and a vector database. By implementing effective strategies to overcome challenges, the project provided a robust solution for processing and querying text from PDF and JSON files. The approach taken and the lessons learned from this project can serve as a valuable reference for future projects involving text processing and conversational AI.