RAG-PDF-Chatbot Project Report

Project Overview

The **RAG-PDF-Chatbot** is an AI-powered **Retrieval-Augmented Generation (RAG) chatbot** designed to answer natural language questions from PDF documents. The system integrates **semantic search** with **vector embeddings** and a **large language model (LLM)** to generate accurate, context-aware responses.

The chatbot allows users to upload one or more PDFs, extracts the textual content, splits it into manageable chunks, and computes embeddings using the **SentenceTransformer** model. When a user asks a question, the system retrieves the most relevant chunks and generates an answer using the **Groq LLM**, ensuring responses are grounded in the uploaded documents.

Key capabilities:

- Upload and process multiple PDF documents
- Extract text and perform semantic search with embeddings
- Retrieve relevant content and generate answers using LLM
- Interactive chat interface with streamed responses via **Gradio**

Enhancements Added

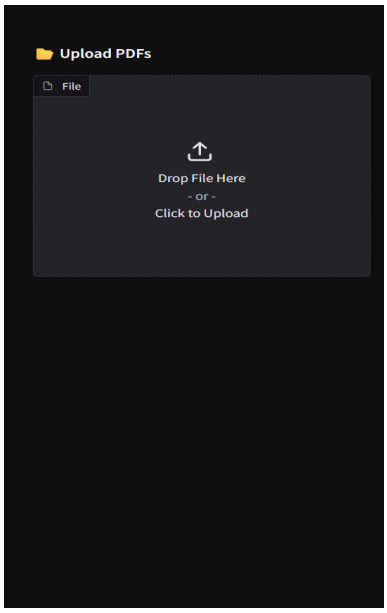
During development, several enhancements were implemented to improve usability and performance:

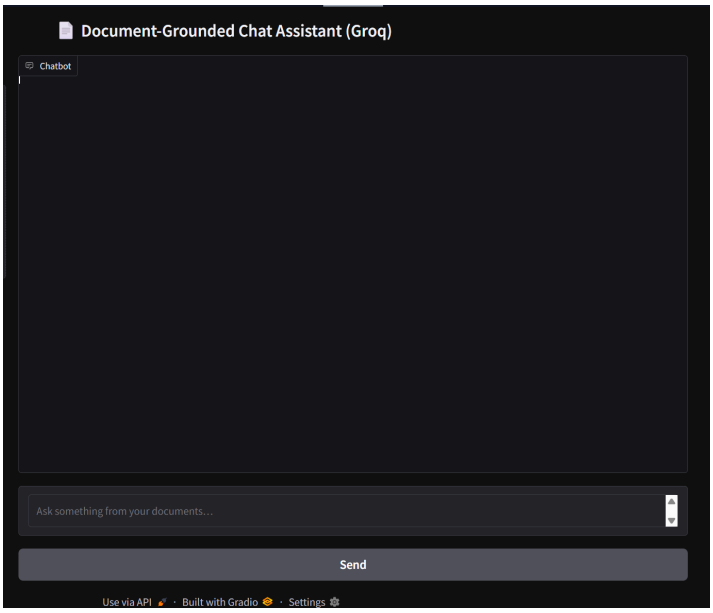
1. **Interactive Chat Interface**
    - Built with **Gradio Blocks**
    - Supports multiple PDF uploads and live streaming of answers
2. **Text Chunking & Embeddings**

- Optimized chunk size (400 words with 50-word overlap) for better semantic search
- Vector similarity using **cosine similarity** to retrieve top relevant chunks

3. **Error Handling & User Feedback**
   - Graceful handling of missing or unreadable PDFs
   - Checks for missing API key with informative error messages
4. **Environment Configuration**
   - API key securely loaded from Hugging Face **Variables**
   - Ensures safe deployment without hardcoding credentials

Screenshots of the Running App





Challenges Faced

1. **Environment Variables & API Key**

- Initially the app crashed due to a missing **GROQ_API_KEY**
- Resolved by securely setting it in Hugging Face Variables

2. **PDF Text Extraction**
    - Some PDFs had pages with unusual formatting, requiring robust error handling
    - Implemented try-except blocks to prevent app crashes
3. **Large Model Loading**
    - LLaMA-3.1 8B model took time to load in Hugging Face Spaces
    - Mitigated by proper caching and optimized pipeline execution
4. **Real-Time Streaming**
    - Implemented character-by-character streaming for better UX, which required careful state handling in Gradio

---

Conclusion

The **RAG-PDF-Chatbot** successfully integrates document retrieval and AI answer generation, providing a user-friendly tool for interacting with PDFs. It is **extensible**, **secure**, and **ready for deployment**, making it suitable for academic, research, and business applications where document-grounded AI responses are needed.