RAG Chatbot Assignment Report

Overview of What Was Built

I developed an intelligent Retrieval-Augmented Generation (RAG) chatbot using Python, Gradio, and a Groq-hosted LLM (llama3-8b-8192). The chatbot allows users to upload multiple PDF files, extract and semantically chunk the text, perform similarity-based retrieval using embeddings, and generate accurate answers from the content via the LLM. The application is deployed on Hugging Face Spaces for public access.

Enhancements Added

1. Sentence-Transformers Embeddings

Instead of basic TF-IDF, the app uses sentence-transformers (all-MiniLM-L6-v2) to generate high-quality semantic embeddings for better retrieval accuracy.

2. Secure API Integration using Hugging Face Secrets

The Groq API key is managed using Hugging Face's Secrets feature, enhancing security and making the app deployment-ready without exposing credentials.

Challenges Faced

- Keras/Transformers Conflict

The latest transformers library wasn't compatible with Keras 3, causing import errors. This was resolved by manually installing tf-keras and resolving version mismatches.

Large File Handling

Processing large PDFs with many pages slowed down chunking and embedding. This was mitigated using RecursiveCharacterTextSplitter with a balance of chunk size and overlap.

- Groq API Key Security

Ensuring the key wasn't hardcoded required learning to use Hugging Face Secrets and .env files effectively.

Hugging Face Space Link

URL: https://huggingface.co/spaces/arslan019/rag-chatbot-pdf