Information Retrieval System using RAG

Task Objective:

Develop a domain-specific Information Retrieval (IR) system with the following key components:

- **Data Layer**: A large, domain-focused corpus stored as vector embeddings within a vector database.

- **Query Intelligence**: A query suggestion module that enhances user input for better retrieval accuracy.

- **Ranking Engine**: Capability to return top-ranked responses with associated relevance metrics.

- **User Interface**: An intuitive front-end interface enabling user interaction and real-time query processing.

- **Implementation Language**: Python.

Implementation Summary:

The Information Retrieval System was implemented using the Retrieval-Augmented Generation (RAG) architecture. The solution integrates a Qdrant vector store to manage and query document embeddings effectively.

System Design:

- Qdrant Vector store is used as the vector database to store document embeddings.

- PDF documents are processed and converted into embeddings using Hugging Face models.

- The embeddings are stored locally in a persistent folder path, accessible by Qdrant.

- A retriever system loads and retrieves relevant chunks of information based on user questions.

- Streamlit is used to create an intuitive front-end interface with two main functionalities:

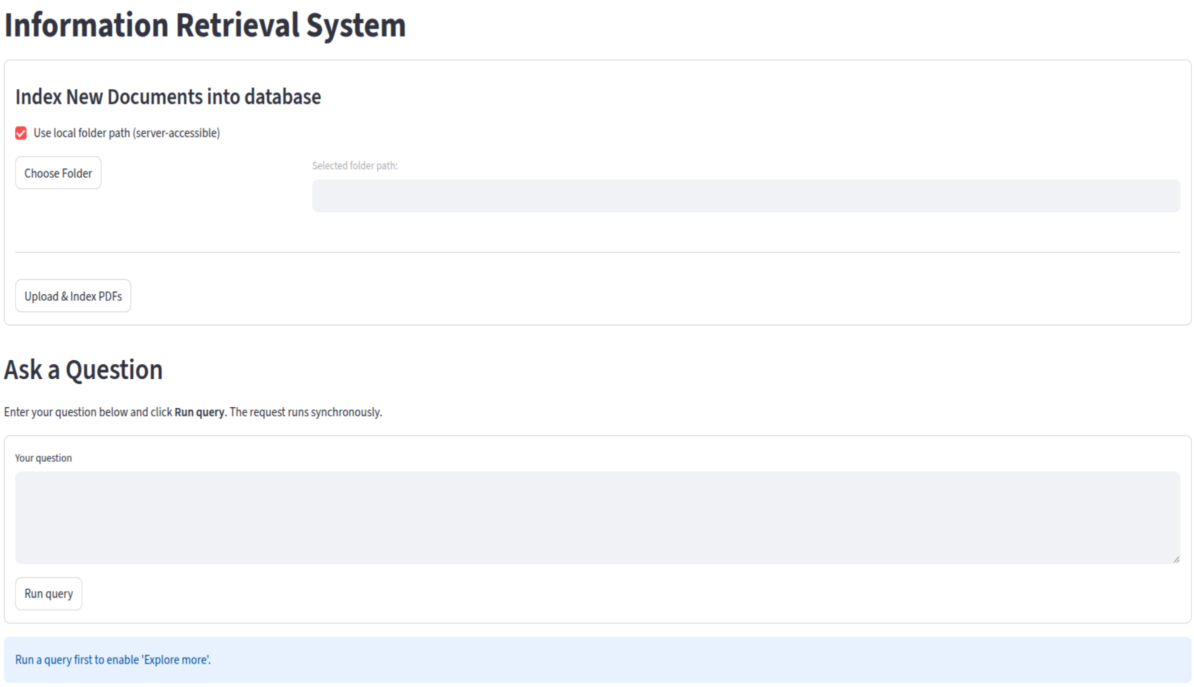
Streamlit UI Features:

1. **Index New Documents** – Allows users to select a local folder path containing PDFs. Upon execution, the documents are read, and their embeddings are stored in the Qdrant vector database.

2. **Ask a Question** – Users can type a question, which is then matched against stored embeddings. The retrieved text segments are displayed as the most relevant responses.

3. **Explore More Option** – Provides detailed exploration of retrieved results, including relevance scores for transparency and evaluation.

The UI for the system is as follows:



Information Retrieval System

Index New Documents into database

☑ Use local folder path (server-accessible)

Ask a Question

Enter your question below and click **Run query**. The request runs synchronously.

Run a query first to enable 'Explore more'.

Conclusion:

This project successfully demonstrates a simplified yet functional Information Retrieval System using the RAG paradigm. It combines document embedding storage, semantic search capabilities, and an interactive Streamlit-based UI. Such systems form the foundation for advanced applications in research, customer support, and knowledge management.