[Siddharth.V,Krishna Pravin,Surya KP,Rijesh Krishna,Gladdin Shruthi]

SRM Institute of Science and Technology
Team:- Knight Rider`s

AI-Powered Research Intelligence[Second Mind]

1. Introduction

This project is an AI-powered **Retrieval-Augmented Generation (RAG) chatbot** designed to enhance research intelligence by efficiently retrieving and synthesizing relevant data. The chatbot leverages **Machine Learning (ML)** and **Natural Language Processing (NLP)** to provide precise, data-backed responses.

2. Core Concepts

2.1 Retrieval-Augmented Generation (RAG)

RAG is a **hybrid AI approach** that combines **retrieval-based search** with **generative AI models** like GPT to generate context-aware responses.

- **Retrieval Phase**: Fetches the most relevant documents using **FAISS vector search**.
- **Augmentation Phase**: Enhances user queries with retrieved context.
- **Generation Phase**: Uses GPT to generate detailed, context-rich answers.

2.2 FAISS - Vector Similarity Search

FAISS (**Facebook AI Similarity Search**) is a powerful tool for **high-dimensional vector search**. It helps in efficiently finding the most relevant research data.

- **Indexing**: Converts text embeddings into a searchable vector index.
- **Querying**: Finds the nearest matching documents based on vector similarity.

Key Code Snippet:

```
import faiss
index = faiss.IndexFlatL2(embedding_size)
index.add(embedded_documents)
```

2.3 FastAPI - High-Performance API Framework

FastAPI is a modern **Python web framework** used to create fast and scalable APIs. It powers the chatbot backend, handling API requests efficiently.

Key Code Snippet:

```
from fastapi import FastAPI
app = FastAPI()

@app.get("/query")
def query_chatbot(user_input: str):
    return generate_response(user_input)
```

3. Implementation Overview

3.1 Data Processing Pipeline

1. **Data Ingestion**: Research papers and documents are preprocessed.
2. **Embedding Creation**: Converts text into numerical vector embeddings using **OpenAI Embeddings API**.
3. **FAISS Indexing**: Stores embeddings in a FAISS index for fast retrieval.
4. **Chatbot Querying**:
    - Converts user queries into vector embeddings.
    - Retrieves top matching documents from FAISS.
    - Uses GPT to generate responses based on retrieved data.

3.2 Key APIs & Endpoints

Endpoint	Method	Description
`/query`	GET	Retrieves an AI-generated response based on a user query.
`/upload_data`	POST	Allows users to upload research papers for indexing.
`/update_index`	POST	Reindexes the FAISS vector database.

4. Future Enhancements

Integrate Multimodal AI: Enhance retrieval by incorporating **images and PDFs**. **Real-Time Learning**: Improve chatbot accuracy through **continuous learning** from interactions. **Multi-Language Support**: Enable NLP capabilities in **multiple languages**.

5. Conclusion

This AI-powered chatbot **revolutionizes knowledge discovery**, making research and data retrieval **faster, smarter, and more accessible**. Its **scalable architecture**, **RAG-based approach**, and **cutting-edge ML techniques** position it as a next-gen tool for research intelligence.