Local Retrieval-Augmented Generation (RAG) with Ollama

Project Overview

This repository is designed for teaching how to set up and run a **Local Retrieval-Augmented Generation (RAG)** system entirely on your laptop. By leveraging the **Ollama** platform, which runs powerful language models locally, you can perform advanced natural language queries on your local data files, all without needing an internet connection. The system is lightweight, making it accessible even on standard laptops, yet powerful enough to handle complex queries and information retrieval tasks.

The primary goal of this project is to help you understand how **RAG** systems work, and how they combine the strengths of **retrieval** (finding relevant data) and **generation** (producing natural language responses). This setup is especially valuable for scenarios where privacy is critical, as all data remains on your local machine.

Key Concepts

1. Retrieval-Augmented Generation (RAG)

RAG enhances large language models by retrieving relevant information from external data sources, such as documents or databases. In this project, we use **embeddings** to represent the meaning of your local documents in a vector space, allowing the model to find relevant information based on the queries you input.

RAG is especially useful when:

- The language model might not have enough context or knowledge about a specific topic.
- The data you're querying is large, but you only need the most relevant portions.
- You need to ensure your queries are answered using up-to-date or specific data from your local files.

2. Embeddings

Embeddings are vector representations of text that capture the semantic meaning of words or documents. By converting your documents into embeddings, we enable the RAG system to quickly and efficiently retrieve the most relevant pieces of information.

In this setup, we use Ollama's `mxbai-embed-large` model to generate high-quality embeddings for your local documents.

3. Local LLMs with Ollama

Ollama is an open-source platform that simplifies running large language models locally on your machine. Instead of relying on cloud-based services, which might raise privacy concerns, Ollama allows you to use powerful LLMs like **Llama3** directly on your laptop.

Setup

This section walks you through setting up the Local RAG system on your laptop. The setup is designed to be straightforward and lightweight, making it ideal for teaching and learning purposes.

Step 1: Clone the Repository

Start by cloning this repository to your local machine:

```
git clone https://github.com/memari-majid/easy-local-rag.git
cd easy-local-rag
Step 2: Install Python Dependencies
Install the required Python packages by running the following command:
```

```
pip install -r requirements.txt
```

These packages include tools for working with language models, document handling, and embeddings.

Step 3: Install Ollama

Ollama is the platform that will allow you to run large language models (LLMs) locally on your laptop. Download and install it from the official website:

Download Ollama Once installed, Ollama will give you access to a variety of language models that are optimized for local use.

Step 4: Pull the Required Models

You need to download specific models that will power the retrieval and generation capabilities of this project. Use the following commands to pull the necessary models:

Llama3: A general-purpose language model for generating natural language responses.

```
ollama pull llama3
```

mxbai-embed-large: A high-quality embeddings model used for retrieving relevant information from your documents.

```
ollama pull mxbai-embed-large
```

These models will be stored locally on your machine, allowing you to run everything offline.

Step 5: Upload Your Documents

The system supports .pdf, .txt, and .json formats for document input. To upload your local documents, run the following script:

```
python upload.py
```

This script processes the documents and generates embeddings for them, making them ready for the RAG system. You can upload multiple files at once, and the embeddings will be stored locally for future queries.

Step 6: Query Your Documents

Once your documents are uploaded, you can start querying them using the RAG system.

With Query Rewriting: If you're asking vague or unclear questions, the system can rewrite your query to improve retrieval accuracy. To run the RAG system with query rewriting, use the following command:

```
python localrag.py
```

Without Query Rewriting: If you prefer to run the system without query rewriting (for more direct control over the results), you can use:

```
python localrag_no_rewrite.py
```