BookFinder-LLM-Application

Overview

BookFinder-LLM-Application is a powerful tool that leverages **Retrieval-Augmented Generation (RAG)** to provide contextual responses based on PDF documents. By utilizing **GPT-2** from HuggingFace as the language model and **Qdrant as the vector database**, this application ensures efficient and accurate semantic search and response generation. The embeddings are generated using the **BAAI/bge-large-en** model from HuggingFace, allowing for nuanced and context-aware interactions with the ingested documents.

Problem Statement

To begin, users input their preferred book genre. The system then retrieves and displays the top 10 books within that genre based on ratings and popularity. Users can browse through summaries and reviews to help them decide on one book. Once a selection is made, the system confirms the choice and provides additional details as required. Finally, the interaction concludes with a courteous thank you message for using the service.

Development Tools

- **LLM**: HuggingFace open-source LLM models
- **Vector Store**: Qdrant for storing and querying book data
- **Embeddings**: HuggingFace BGE Embeddings for semantic search
- **Web Framework**: Streamlit for the user interface

Approach and Reasoning
Data Ingestion

1. **Document Loading**: The PyPDFLoader is used to load the book data from a PDF file.
2. **Text Splitting**: The RecursiveCharacterTextSplitter splits the documents into manageable chunks for better processing.
3. **Embedding Model**: The HuggingFaceBgeEmbeddings model encodes the text chunks into embeddings for semantic search.
4. **Qdrant Vector Store**: The embeddings are stored in Qdrant, a vector store, to enable efficient similarity searches.

Application Workflow

1. **Initialization**:
   - The embeddings model (**BAAI/bge-large-en**) and the **Qdrant** client are initialized.
   - The Qdrant database is set up to manage the book data.
2. **User Interface**:
   - **Home Page**: Provides an overview and navigation options.
   - **Generate Response Page**:
     - The agent greets the user and awaits input on the desired book genre.
     - Upon receiving the user's request, a semantic search is performed on the Qdrant database to retrieve relevant book data.
     - The agent uses the retrieved context to generate a response with the help of the **Hugging face GPT-2** model.
   - **Contact Us Page**: Provides a form for users to contact the developer.
3. **Semantic Search and Response Generation**:
   - The semantic search retrieves the top 5 relevant book documents based on the user's query.
   - The Hugging face GPT-2 model uses the retrieved context to generate a detailed response.
   - The user is guided through narrowing down the selection to one book and receives a thank you message upon task completion.

Features

- **PDF Document Ingestion:** Load and split PDF documents into chunks for semantic search.
- **Embedding Generation:** Use HuggingFace's BAAI/bge-large-en model for generating embeddings.
- **Contextual Search:** Perform semantic searches to retrieve relevant document sections.
- **Response Generation:** Use GPT-2 from HuggingFace to generate responses based on the retrieved context.
- **Streamlit Interface:** User-friendly interface with navigation options for Home, Generate Response, and Contact Us pages.

Harsh Kumawat 9928967278