Article Retrieval System - Report

Karolina Surówka

May 4, 2024

1 Introduction

This report presents the process of the development and implementation of an Article Retrieval System leveraging state-of-the-art techniques in natural language processing and information retrieval. The system is designed to index and retrieve fragments from articles contained within the "1300 Towards Data Science Medium Articles" dataset sourced from Kaggle.

The core components of the system include an innovative indexing strategy that enables efficient searching of articles, and a Retrieval Augmented Generation (RAG) mechanism powered by the Cohere platform, Chroma, and the LLM model.

2 Strategy

The strategy of this project revolves around leveraging cutting-edge techniques in natural language processing (NLP) and information retrieval to create an efficient and effective Article Retrieval System. Below is presented a breakdown of the key components of the strategy:

- **Data Acquisition and Preparation**

  The project starts with obtaining the dataset containing Medium articles from the "1300 Towards Data Science" collection available on Kaggle. This dataset serves as the foundation for building the retrieval system.

- **Indexing Strategy**

  An essential aspect of the strategy is devising an indexing strategy that allows for efficient searching of articles. This involves structuring the data in a way that facilitates quick access and retrieval based on user queries. In this project, DataFrameLoader is used to load the dataset, and RecursiveCharacterTextSplitter is employed to chunk the articles into manageable fragments for indexing.

- **Retrieval Mechanism**

  The heart of the system lies in its retrieval mechanism, which employs advanced techniques such as Retrieval Augmented Generation (RAG) powered by Cohere. This mechanism enables the system to find and return relevant fragments from articles in response to user queries. The use of Cohere's embeddings and Chroma allows for semantic similarity search, enhancing the relevance of retrieved fragments.

- **Chunking Strategy**

  A thoughtful approach to chunking is adopted to strike a balance between fragment length and content richness. By determining appropriate chunk size and overlap, the system ensures that retrieved fragments contain relevant information while remaining concise.

- **Integration of Conversational AI**

  To enhance user interaction and provide more informative responses, the system integrates a Conversational AI component powered by LLM model. This allows users to engage in natural language conversations with the system, further improving the user experience. By leveraging the LLM model "llama2" from OpenAI, the system can generate contextually relevant responses to user queries.

3 Selected Technologies

- **LangChain**, a comprehensive framework designed for developing applications that leverage the capabilities of large language models (LLMs). This framework provides developers with the tools and resources needed to build powerful and innovative applications that harness the potential of cutting-edge language technologies.
- **Cohere**, a sophisticated platform that specializes in extracting embeddings from text data. These embeddings capture semantic information about the underlying text, facilitating tasks such as similarity search, clustering, and classification.
- **Chroma**, an AI-native open-source vector database developed by LangChain. It is specifically designed to store and query vector embeddings efficiently, enabling fast and scalable similarity search operations. Chroma is optimized for handling high-dimensional vector data generated by LLMs and other NLP models, making it an ideal choice for applications that require complex semantic search capabilities
- **Ollama**, a conversational AI model developed by LangChain, with the **"llama2"** variant being one of its implementations.

4 Encountered Challenges

Undeniably this whole project was a challenge for me, because of its complexity and many new concepts, technologies that I have never used before. Nevertheless while creating Article Retrieval System I was absorbed in curiosity about this topic and, despite the difficulties, I wrote it with great motivation and determination.

The biggest difficulty for me was choosing technologies, of which there are plenty. I spent a lot of time reading articles and blogs devoted to this topic and testing various technologies and models. The challenges I encountered were various limitations of the technologies used, such as API rate limits, memory availability, and adapting the appropriate data format.

5 Areas For Future Development

- **Optimization:** Further optimization of the system's indexing and retrieval processes can be pursued to improve overall performance, including reducing response times for query processing and enhancing scalability to handle larger datasets. Optimization for scalability and efficiency could enable robust performance across diverse deployment environments.
- **Enhanced Semantic Understanding:** Advancements in natural language processing techniques can deepen the system's grasp of text semantics, refining interpretation and response capabilities.
- **Fine-tuning Retrieval Mechanisms:** Continuously fine-tuning the retrieval mechanisms, such as RAG, to improve the relevance and accuracy of retrieved article fragments in response to user queries. This can involve experimenting with different embedding models, tuning hyperparameters, and incorporating user feedback into the ranking algorithms.