Implementación de un Sistema RAG sobre PubMed

Brais Bea Mascato

Introducción

Este proyecto tiene como objetivo desarrollar un sistema de Recuperación Aumentada con Generación (RAG) que consulta PubMed, extrae los primeros cinco resultados de la base de datos PMC (artículos de acceso abierto), realiza chunking y embedding de esos artículos, y almacena los embeddings en un índice (FAISS). Luego, el sistema busca los fragmentos más relevantes para responder la consulta del usuario utilizando un modelo de lenguaje grande (LLM). Finalmente, almacena la consulta, la respuesta generada y los extractos utilizados como contexto en una base de datos.

Tecnologías Utilizadas

- **Python** como lenguaje de programación.
- **FAISS** para la indexación y búsqueda de embeddings.
- **Hugging Face Transformers** para generar embeddings.
- **LangChain** para gestionar el chunking y el embedding.
- **PyMed** para realizar consultas a PubMed y obtener artículos completos.
- **SQLite** para almacenar las consultas y respuestas generadas.

Pipeline del Sistema

1. El usuario formula una pregunta.
2. Una primera llamada a un LLM convierte la pregunta a una consulta avanzada de PubMed en inglés y en formato graphQL.
3. La consulta se usa para obtener cinco artículos open-access de PMC.
4. Se realiza chunking y embedding de los artículos.
5. Se almacenan los embeddings en FAISS.
6. Se buscan los cinco fragmentos más relevantes en FAISS para responder la consulta.
7. Se genera la respuesta final usando un LLM con los fragmentos recuperados como contexto.
8. Se almacena la consulta, la respuesta generada y los extractos utilizados en una base de datos SQLite.

***Nota:** En caso de que no se pueda traer el artículo completo en el punto 4) se hará embedding de todos los abstracts de la llamada a la API y se traerán 5 abstracts con los embeddings más similares a la pregunta del usuario.

Implementación

El sistema se estructura en cuatro clases principales:

1. `ModelManager`: Inicializa el modelo LLM y genera consultas avanzadas y respuestas.
2. `PubMedRetriever`: Maneja la consulta a PubMed y obtiene los artículos relevantes.
3. `EmbeddingProcessor`: Gestiona el chunking y embedding de los textos.
4. `DatabaseManager`: Almacena consultas, respuestas y contexto en una base de datos postgresSQL.

Diseño del workflow: