

Informe RAG:

Tecnicatura Universitaria en Inteligencia Artificial

Fecha: 27/02/2024

Integrante:

- Mateo Rovere

Profesor:

- Juan Pablo Manson
- Alan Geary
- Andrea Carolina Leon Cavallo
- Ariel D'Alessandro

Se puede encontrar todo el repositorio en mi github:

https://github.com/Mateorovere/TP2_NLP

RAG (Retrieval-Augmented Generation):

Introducción:

RAG (Generador Aumentado por Recuperación) es un modelo de lenguaje de última generación que combina la potencia de la generación de lenguaje con la precisión de la recuperación de información.

¿Cómo funciona?

1. **Entrada:** Se le proporciona al modelo una consulta o un texto corto como entrada.
2. **Recuperación:** RAG busca en una gran base de datos de documentos relevantes para la consulta.
3. **Generación:** El modelo utiliza la información recuperada para generar un texto nuevo y original que responde a la consulta.

Ventajas de RAG:

- **Precisión:** La información recuperada por RAG ayuda a garantizar que el texto generado sea preciso y relevante para la consulta.
- **Creatividad:** RAG puede generar textos nuevos y originales que van más allá de la simple recuperación de información.
- **Versatilidad:** RAG se puede utilizar para una amplia gama de tareas, como la generación de resúmenes, la traducción automática, la escritura creativa y la respuesta a preguntas.

RAG es un modelo de lenguaje poderoso y versátil que tiene el potencial de revolucionar la forma en que interactuamos con la información.

Yo elegí implementar RAG que sea experto sobre la anatomía humana a partir de libros de fuentes confiables, a partir de un archivo csv (que contiene información de los sistemas del cuerpo) y de datos de wikidata.

Para hacer split en los textos use RecursiveCharacterTextSplitter con un chunk_size de 500.

Teniendo en cuenta que el entorno de Colab tiene como límite 13 mil millones de parámetros para los modelos, elegí a la versión de 13b de LLAMA 2, dado

que cumplía ese requisito y que tenía buen performance en el lenguaje natural.

El modelo de embedding que elegí es el “intfloat/multilingual-e5-base”, dado que la versión “large” del mismo me daba error por CUDA debido a un OOM por el colab.

Luego a las bases de datos vectoriales estaban con chromaDB