Trabajo práctico final

Procesamiento del lenguaje natural

Alumno: Max eder

1)

Introducción:

Para el trabajo final de la materia, se requería la construcción de un chatbot experto en un tema a elección, utilizando la técnica RAG (Retrieval-Augmented Generation), y que incluyera especificaciones sobre dónde extraer el contexto para agregarlo a un modelo de LLM. Comencé con la elección del tema y, posteriormente, con la búsqueda de datos para alimentar las diferentes bases. Luego, fue un desafío organizar el orden en el que iba a llevar a cabo el proyecto.

El tema que elegí fue la Segunda Guerra Mundial. Para ello, utilicé tres archivos PDF que suman aproximadamente 250 páginas, los cuales relatan hechos y brindan un marco histórico sobre la guerra. Además, empleé un archivo CSV que contiene una columna con eventos y otra con las fechas en que ocurrieron. Para la base de datos de grafos, decidí hacer consultas a Wikidata, perfilando estas consultas cuando se trataba de personajes importantes de la guerra.

Desarrollo:

Comencé trabajando con los archivos PDF, los cuales traje desde una carpeta de Google Drive. Luego, empecé a procesarlos realizando el chunkeado (fragmentación) con la herramienta CharacterTextSplitter de Langchain. A continuación, limpié los textos para eliminar caracteres innecesarios y realicé los embeddings, guardándolos en la base de datos vectorial. Finalmente, armé una función que retorna el contexto, devolviendo el chunk más cercano a la consulta.

En el caso del archivo .csv, lo traté extrayendo las ubicaciones o eventos de las consultas utilizando la técnica NER de Spacy. Luego, traduje los resultados al inglés para que coincidieran con las filas del archivo, ya que estaba en ese idioma.
En las consultas a Wikidata, utilice la misma herramienta para extraer el nombre del personaje.
Una vez que tenía listas las consultas a las tres bases de datos, me dispuse a crear los clasificadores. Opté por quedarme con el de ejemplos y comparación de embeddings porque resultaba ser más eficiente.
Por último, utilicé el modelo de LLM 'zephyr' de Hugging Face. Agregando el contexto que, luego de ser clasificado, me devolvían las funciones, armé el chatbot con la interfaz de Gradio.

Conclusión:

Si bien creo que logré que el chatbot responda de forma dinámica las consultas, basándose en el contexto que se agrega, noto que podría tener algunas fallas en ciertas cuestiones.

Por ejemplo, si se hace una pregunta que se perfila para buscar en los PDF y esta información no se encuentra, el modelo igual va a devolver un chunk por proximidad que probablemente no tenga relación con la información que se desea obtener.

En el caso de los clasificadores, creo que podría haber hecho algo más robusto, con otra ingeniería de prompts para el caso de Zero Shot, o generar un modelo de clasificación con más ejemplos en el caso de los embeddings.

En definitiva, con más tiempo, pienso que se podrían pulir algunos errores, realizando más pruebas o alimentándolo con más datos.

2)

Rerank es una herramienta que mejora la calidad de los resultados de búsqueda reorganizando los documentos recuperados inicialmente para que los más relevantes aparezcan primero. Esto es útil porque asegura que los resultados más importantes estén en la parte superior, minimiza la aparición de resultados no relacionados y ahorra tiempo y recursos. Funciona dividiendo la información en partes más pequeñas y detalladas, utilizando métodos avanzados para mejorar la coincidencia de palabras clave y contexto, y generalizando mejor las nuevas consultas y documentos, mejorando así la calidad de la búsqueda. Este proceso mejora la precisión y la experiencia del usuario al reducir el ruido y proporcionar respuestas más claras y concisas. En mi proyecto, aplicaría Rerank después de obtener el contexto de las bases de datos (PDF, CSV, Wikidata) y antes de generar la respuesta final con el modelo LLM. La integración del Rerank se haría justo después de la función que obtiene el contexto de las diferentes fuentes de datos y antes de preparar el prompt y generar la respuesta final. Esto aseguraría que los resultados más relevantes se utilicen para construir la respuesta, mejorando así la efectividad del chatbot.

A[Consulta del Usuario] --> B[Recuperación Inicial de Documentos]

B --> C[Lista Inicial de Resultados]

C --> D[Rerank]

D --> E[Lista Reordenada de Resultados]

E --> F[Generación de Respuesta Final]

Mastering RAG: How to Select A Reranking Model - Galileo (rungalileo.io)