Proyecto de Generación de IA

Chatbot simple en ambiente local usando un modelo de generación de texto

Por: Fausto Morales
ffmogbaj@gmail.com

1. Propuesta

La propuesta de proyecto final consistió en realizar un Chatbot ejecutado en un entorno local empleando mi computadora personal MAC M2 usando un modelo de generación de texto como Deepseek.

Se propuso integrar por lo menos la arquitectura de front, APIS y el módelo. Y se comentó como opcional realizar un Fine Tunning, sin embargo se observó que los recursos para realizar esta actividad son elevados.

2. Propuesta final

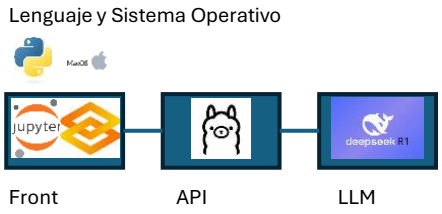
Derivado al comentario de dificultad para un Fine Tunning se descartó la opción y en su lugar se exploró la generación aumentada por recuperación (RAG) que extiende las capacidades de los LLM a una base de conocimientos adicional.

Para ello se realizaron 3 casos:

1. Chat bot incluyendo sólo LLM cumpliendo el requerimiento mínimo de la propuesta.
2. Chat bot incluyendo LLM + RAG con un archivo pdf.
3. Chat bot incluyendo LLM + RAG con una carpeta de archivos pdf.

El diagrama de esta solución quedó de la siguiente forma:

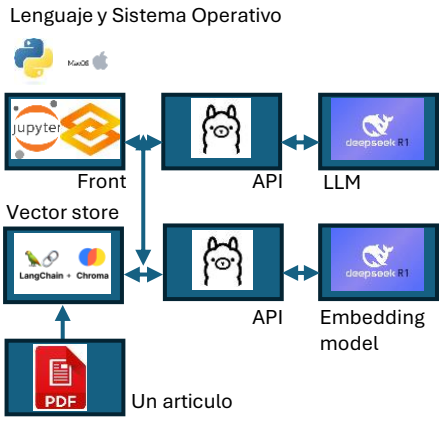
1. Chat bot incluyendo sólo LLM.



Se empleó una Macbook Air M2 con python3 dentro de jupyther. Este prototipo se corrió completamente en jupyther y el notebook sirve de front para su interacción con el usuario empleando el paquete: gradio.

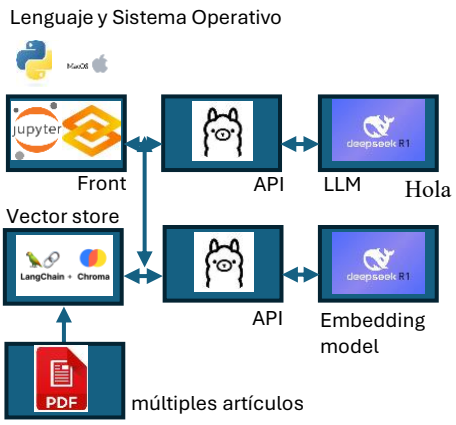
Para poder interactuar con el modelo se empleó el software y APIS locales instalando Ollama que permitieron descargar e interactuar con el módelo más pequeño de Deepseek: "deepseek-r1:1.5b"

2. Chat bot incluyendo LLM + RAG con un archivo pdf.



Para este nuevo modelo se integró el RAG a partir de un artículo de Wikipedia .pdf el cúal se transformó y almacenó temporalmente en embeddings (o almacenamiento de vectores) a través de LangChain y Chroma, en conjunto con el API de Ollama y nuevamente DeepSeek: "deepseek-r1:1.5b" como módelo de Embeddings.

3. Chat bot incluyendo LLM + RAG con una carpeta de archivos.



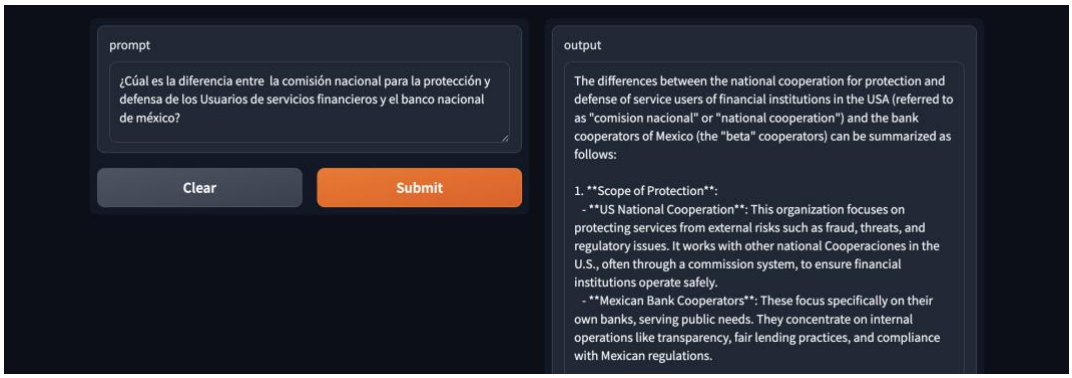
Para esta última versión del modelo se creó una carpeta de archivos para nutrir el almacenamiento de vectores y al mismo tiempo se almacenó este elemento Chroma en la carpeta local: ./chroma

Nota: Como pre-requisito para el funcionamiento correcto de estos componentes fue necesario instalar con brew: poppler y tesseract, así como diferentes dependencias Python.

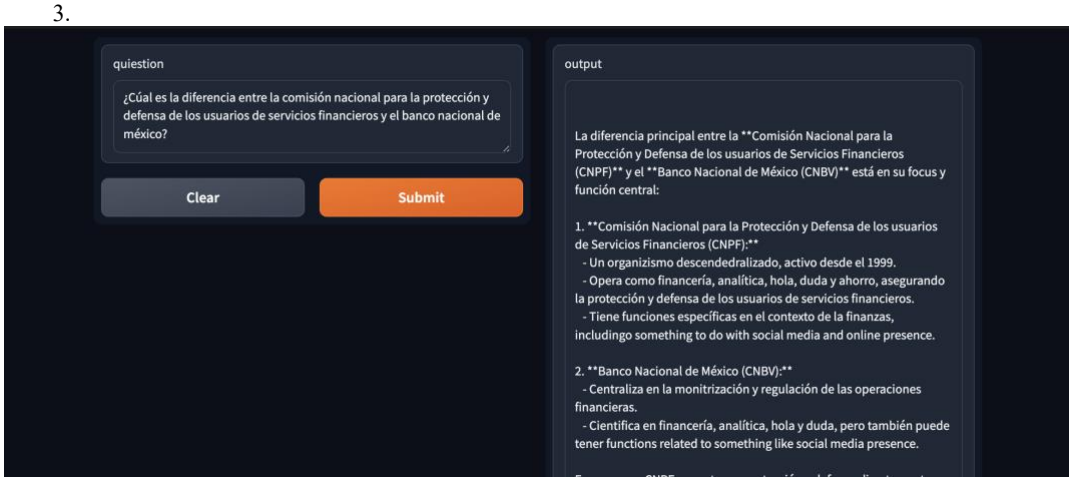
3. Resultados

Para poder testear los chatbots se les preguntó a los 3 modelos la misma pregunta: "¿Cuál es la diferencia entre la comisión nacional para la protección y defensa de los usuarios de servicios financieros y el banco nacional de México?".

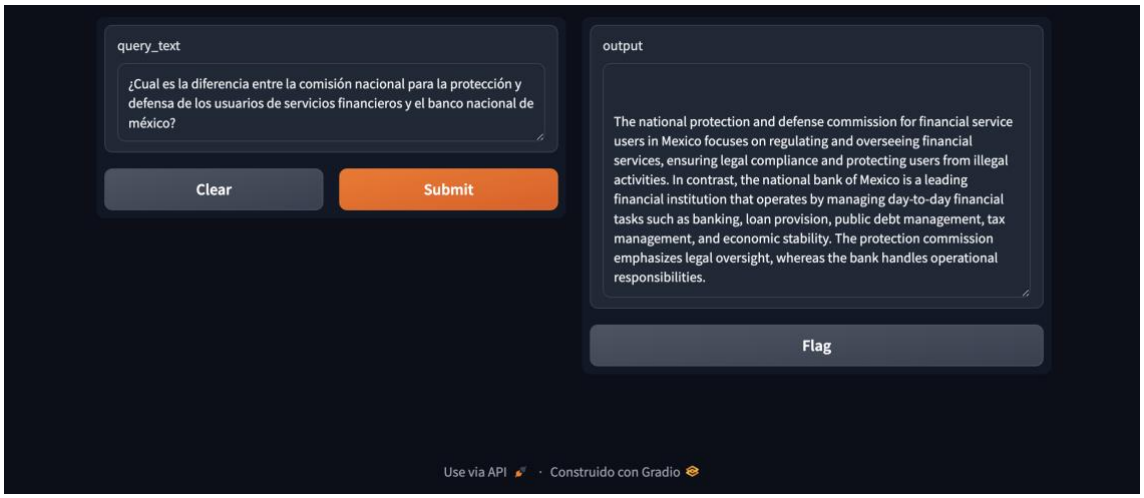
1. Para el módelo 1, la respuesta es incorrecta y se encuentra que el modelo original no tenía información correcta de esta pregunta:



2. Para el módelo 2, se empleó la página de la wikipedia de condusef. Esté fue su respuesta, también errónea:
3.



4. Por último para el módelo 3, se empleó la página de la wikipedia de condusef, la página de wikipedia de banamex y una página aleatoria del banco de México sobre el Artículo 48 Bis 2 de la Ley de Instituciones de Crédito. Siendo correcta su respuesta:



Nota, aunque a veces la respuesta era incorrecta ya que existían variaciones si se volvía a preguntar

4. Conclusiones

Fue posible realizar una demo de: un chatbot simple en ambiente local empleando un modelo de generación de texto por LLM (Large Language Model o Modelo de Lenguaje de Gran Escala) a partir de tecnologías que uso por primera vez como oLlama y LangChain, sin embargo, requiere mejoras para poder aplicarlo en entornos productivos y empresariales cómo, por ejemplo:

- Lograr tener una base de conocimientos especializada más amplia y optimizaciones para escoger información relevante.
- Mejorar el entorno de ejecución del modelo, implementando un módulo óptimo, así como sistemas frontales y de APIS especializados y dedicados con seguridad productiva.
- Realizar monitores sobre las respuestas automáticas generadas, así como una gestión de las respuestas incorrectas.

Puedes consultar el notebook completo en: chatbot.ipynb

5. Referencias

Referencias para este proyecto:
https://aws.amazon.com/es/what-is/retrieval-augmented-generation/
https://www.datacamp.com/es/tutorial/deepseek-r1-ollama
https://github.com/tonykipkemboi/ollama_pdf_rag/blob/main/notebooks/experiments/updated_rag_notebook.ipynb
https://github.com/pixegami/rag-tutorial-v2/tree/main
https://github.com/fcori47/rag_basico/blob/master/Clase%206%20-%20Final.ipynb
https://aws.amazon.com/es/what-is/retrieval-augmented-generation/
DeepSeek-AI (2025) presenta *DeepSeek-R1: Incentivizing Reason zng Capability in LLMs via Reinforcement Learning* ([arXiv:2501.12948](https://arxiv.org/abs/2501.12948)).