Name: Yang Sen Wei, Aaron

Response to Statement 3 part 3

**Question 1**: Please make use of free/open source models (i.e. ChatGPT free plan) and note down any limitations you face using this methods

**Response:**

I am very appreciative of the open source models such as Llama models and many other popular open source models. These models would be very capable as well even if these are free and open-source compared to proprietary models.

However, the limitation that I had encountered would be the integration of the HuggingFace open source models into Langchain. The original intent is to use LLM models from HuggingFace and Langchain integration, allowing for a smoother and more straightforward integration. Unfortunately, the response generated by the model was not as intended and after some checking, I had realized that the HuggingFacePipeline API has some bugs that were reported on Github as well. It was suspected that the prompts are not customized to the LLM specific prompt formats, which resulted in not as intended outputs.

Nonetheless, I was able to use ollama for integration with Langchain by downloading locally the Llama 3 instruct model and the LLM was able to generate the appropriate responses to the question of “what are the differences between the Swiss and UK acts” as well as some other test questions.

**Question 2**: Alternatively explain in detail how you would make used of paid third-party APIs if made available to you.

**Response:**

If I had make used of paid third-party APIs if made available to me, I would have trialed the following

1. Vector Store – I would trial uploading the embeddings to a cloud vector store such as Azure Blob Storage, Pinecone or Weaviate for persistence. I had implemented Chroma as a vector db. However, I believe that by uploading the embeddings to a cloud service, it can aid in scaling the operations such as using RAG for question-answering. That said, Chroma is in the works to release a cloud vector database and I am extremely excited about the upcoming feature.
2. Prompt Engineering – The process for the prompt tweaking presently can be considered to be more manual. As such, it would be good to have access to some third party APIs like Azure Prompt Flow or Langsmith, which would allow scaling of prompt comparisons as well as having a stateful log of the prompts and outputs.
3. LLM – It would be interesting to trial the latest GPT models as GPT is one of the top models for quality in generation of text. As such, it would be interesting to observe how it compares to llama 3 that was used in this assignment. Having access to LLM endpoint such as HuggingFace may have mitigated the problem of the HuggingFacePipeline API earlier as well. Third-party APIs such as AWS S3 or Azure can allow hosting of models for serverless inference (MAAS), allowing for better scaling and deployment as well.