Justin Lim
CS224G

Insights From Building DiligenceDynamics: an AI-powered Web App

Throughout working on our project, DiligenceDynamics, I have learned a lot about structuring the architecture behind our conversational question-answering bot, prompt-tuning, general backend architecture for web apps, and working with AI frameworks like LangChain and Pinecone.

One key takeaway was the significance of contextualized questions in making our RAG Q&A bot a more human-like conversation between users and AI. An important extension that we added to our project during Sprint 2 was to use previous chat history to re-contextualize queries. For example, without contextualization, if I asked "What was the reported revenue" and followed up with "How was it segmented", the bot would not understand what "it" refers to in the second query. Thus, we added a second GPT-4 call that takes the previous chat history and uses it to reformulate the question asked by the user depending on what was previously asked. This was a non-trivial task that also required us to iterate on our prompt. Specifically, we had to change the prompt numerous times to make the bot only output the re-contextualized question and nothing else. In the end, we ended up adding a line in the prompt that says "Do NOT answer the question, just reformulate it if needed and otherwise return it as is". Overall, this addition made our RAG model much more conversational and user-friendly, which was important for our overall web app experience.

Another learning through our exploration of prompt-tuning was the importance of few-shot learning. For our investment report copilot, we quickly realized that our original approach of finetuning our model on real investment reports would be infeasible, as creating a curated set of examples to finetune on would be too time-consuming. Thus, we switched our approach to few-shot learning, which proved to be quite effective. In particular, we were able to carefully hand-pick a short yet informative example of an investment report. For me, this showed that we don't necessarily need to go with the theoretically best option of finetuning, but instead put time into prompt-tuning. Given more time, I think we could have done even more with chain-of-thought prompting, self-consistency, and rewording of the prompt to have the investment report give more detail in each paragraph, which is an issue that pretty much all LLMs have.

Another big takeaway from this project was how to design and build a functional backend for a web app. If I were to do this project again, I would have first started with a more solid plan for the backend structure. Initially, we started the project by using Pinecone for our vectors and Firebase for file storage, keeping track of text chunks, and storing chat history. However, once we added in authentication and the notion of users, we had to restructure the entire backend. It was a huge pain to start re-organizing all the data to keep metadata containing the user id such that our APIs would only return data for the authenticated user.

Finally, throughout the coding process for our RAG model, we ran into a ton of issues with using LangChain and Pinecone. From my perspective, these libraries are quite new and change very frequently with all the developments in AI. Thus, the documentation was quite unstable and made it difficult for us to get the code working without issues. For example, LangChain builds its own interface for vector stores like Pinecone, but there isn't any documentation on how to use arguments for the native Pinecone libraries through LangChain. In particular, it was challenging for me to figure out how to use namespaces in Pinecone and pass in custom metadata because it was not documented. Thus, my takeaway from working with these libraries is that they are not always easy to use, so making very specialized solutions can be a much bigger time-sink than expected.

Ultimately, the journey through building DiligenceDynamics has expanded my technical proficiency and underscored the pivotal role of providing context for LLMs, carefully tuning prompts, and carefully designing and building a production infrastructure to power a web app, especially when AI packages are involved. Moving forward, I am excited to apply these insights to future projects for both personal and professional endeavors.