CS 224G Individual Report
Krrish Chawla

Over the course, I finally had the opportunity to work on something with real world impact. For many years, I have been passionate about building things for impact, and AI as a toolkit has opened many dimensions of opportunity. I am a firm believer that AI has three avenues - creativity, automation and decision making. Creativity is awesome, and has a great wow-factor amongst people, but it serves no true purpose for the time being. Automation is very useful and saves a lot of time for people to do their daily tasks, like meeting scheduling, or any other role that an AI could take on. However, where the true potential lies, is decision making. Now we are getting close to AGI, where an artificial being is actually making decisions in real time - decisions that impact life. Thus, for my project for this class, I wanted to explore the avenue of decision making, powered by the automation capabilities of AI. My project team, SkillSyncer, offered me the best position to do this. My teammates and I experienced the same problems in the workspace - getting matched to teams at internships which we didn't care about or were not very skilled at. Further, while working at small startups, I noticed founders having a hard time allocating human resources to different broad ended projects. That's where SkillSyncer enters. It is truly difficult to do for a human since it is not easy to remember who has what experience, how many years, exactly working on what etc, however an AI can remember all this information, and make inferences from it. After a lot of experimentation, failure and learning, we very carefully designed the SkillSyncer pipeline. The final pipeline leverages natural language processing techniques to extract and analyze key information from employee resumes and project descriptions. Text embeddings are used to calculate semantic similarity, and an LLM assists in ranking the best-fit employees for a given project, providing reasoning for its choices. The system dynamically updates the recommendations or the state of the application as employees are added or removed. Further, data security is maintained by masking resumes before passing them into the LLM for running any inference. Every component serves a purpose. Since LLMs are slow, we had to come up with a way to speed up the whole process. Since the number of employees and projects could scale to very high numbers, we used a fast, textual embedding model to shortlist a few top picks. However, we realized that textual embeddings are not very good at reasoning, so it just formed feeder data to pass to the LLM to make the final decisions. In one of the lectures, I learnt about the problem of hallucination. This is a big bottleneck for the decision making aspect of LLMs, since those decisions hold actual weight. To reduce hallucinations, we used the embeddings to provide better, more concise but enough context for the LLM to make a strong reasoning. This served as a different approach to fine tuning, rather than showing the LLM more examples, we showed it a smaller, but more impactful prompt with clear decisions to be made. Also, masking the resumes using the LLM was also an interesting approach. The goal was to not show the LLM data which correlates personal identities to projects. Thus, we extracted the chunk of the resume with personal details, passed that into the LLM to mask it, and then replaced it back in the resume, successfully isolating the personal identities from the work experiences. Prompting was another avenue where I learnt a lot, optimizing prompts with every iteration of the app. All these experiences led me to further believe that while LLMs are sophisticated, they remain tools - the way we communicate our intent to them is undeniably the lever that produces exceptional results.