

CS224G Individual Report
Roy Yuan

I'm glad I was able to take this course and work on a project I thought was actually interesting with my teammates. Although I have done many coding-related projects in the past, most have only existed locally on my computer without having any sort of practical use case. The ability to solely focus on the project without having supplementary assignments gave me a sense of freedom I craved in upper level classes, as I got to work on something I enjoyed rather than required since I got to choose the project. Although I think that many cool things can be done with AI, I believe the best applications of AI are ones that have some sort of tangible impact and can be used in a practical way. Additionally, I believe AI can be applied in a creative way to automate some tedious decision-making processes, to save humans time with such monotonous things so they can focus their efforts on other tasks that AI cannot yet do. While automation for tasks has existed in the past, however, with the introductions of LLMs, that same process can now be done with decision making, which I find innovative and exciting because it means tasks that require some thinking by humans can now also be automated. When I was brainstorming projects and getting ready for our proposal, my team members and I had an experience that we could relate to: work experience at our previous internships. All of us had been assigned to projects with little to no knowledge about why we were assigned to such project. As an example, for myself, at one of my internships I was assigned to read due diligence reports that required very little expertise, which I wondered about given most of my experiences centered around tech. As a result, we decided to build SkillSyncer, for practical use cases based on our previous dissatisfactions. One of the biggest insights I gained during this class was to truly see the efforts it takes to build something from near scratch. Having never done a hackathon before and usually spacing out my work between days, in CS224G, it was different. Many nights my team and I would be unable to actually code because so much time was devoted into discussing what features to implement and getting feedback on them. I have never stayed up as late nor spent as much time on one class, but I enjoyed all of it since I was building something I thought was cool. For the first sprint, I took a bit of time to get adjusted to the work pace but by the 2nd and 3rd sprint, I was able to iterate much quicker and efficiently in our high-paced atmosphere. It was a bit much for me to handle in the beginning, but I thought it was helpful for personal growth as I envision many projects I'll tackle in the future will be similar. My teammates were also very understanding with my weird work hours due to my schedule with the men's basketball team, so I appreciate them a lot for that. I thought the hardest part was getting started with the project, because we had little validation outside of our own experiences to know what to build. I think for myself I was a bit slow to get started, but once we had a functioning model and web skeleton, I was able to develop on top of that much more efficiently while incorporating user feedback into it. For myself, I learned that getting started and building on top is much better than taking forever to start making something. I also enjoyed the problem solving aspect of this project, as we tackled a lot of them spontaneously. When I implemented the LLM to more intelligently select candidates, it slightly brushed away the need for embeddings. However, one issue I ran into was that using the LLM calls was slow, and would be too slow for large databases of resumes/projects so running the LLM on each individual project/resume would not work. After discussing and brainstorming with the team, we

reintroduced the embedding into our environment to cut down the candidate list, and then used the LLM to find the best candidate out of good candidates. For our LLM, I learned that sometimes, you have to treat it like a child for the best results (such as explicitly telling it what to output, how to interpret something you're inputting, etc). By this I mean extremely clear prompting with some good amount of repetition. Without it, the responses were a bit too broad or creative. I personally found that better prompting was more effective than altering temperature score, which affected the creativity of the model (in my experience). I think the most helpful advice I got was the guidance I got from my teammates in the very beginning, given I had little hackathon experience but they did; and another really helpful piece of advice was John's prompting advice, which made me come to some of the conclusions I mentioned above (and helped me get better results). It was only a 10 week class but I feel like I learned more in this time period than any other class, just because of how invested I was in this project because I thought it was a really interesting practical application. I hope to continue exploring LLMs and better understanding what they can be best used for. I'm glad I took this class and I'm even happier that I could work with both of my great teammates.