CSCE 240 Final Project

Jacob Stoll

For this project we were to combine all of our previous chatbot assignments to give a functional final product. The final chatbot should be able to retrieve financial information from two companies and then report on the information they contain. It should also handle any user input and give an appropriate response. I chose Campbell Soup and Coinbase as my two companies, and would retrieve their financial information from their company form 10-Ks. The language I chose to use throughout the project was Python, as I had some basic experience in it and wanted to use it more. Python also made some things easier due to its simpler syntax and ease in writing. I haven't done anything similar to this project before but have had some experience in file parsing which was used throughout the project.

The scope of the project was my two selected companies, Campbell Soup and Coinbase. As for the design and components:

Project 1:

For my project 1, I copied over the contents of the 10-K file for each company and then would parse through it line by line to get the amount of parts, lines, words, etc. However, in my final chatbot, I reused Tarun's code as it provided webscraper functionality. His code allowed a company's link to be inserted and then from there it would grab the company's 10-K file. The difficulty in reuse was not high, though I had to make some slight modifications such as adjusting the starting line that is scraped in order to integrate it with my code.

Project 2:

For my project 2, I created a list of RegExs that would parse a user's input for certain keywords to match with the parts of the 10-K file. If a keyword matched, then that part would be printed to an output file. I also added RegExs that prevented single numbers to be printed or

“Table of Contents” as they were repeated throughout the 10-K due to formatting issues when copying. Random text as an input was accounted for as it wouldn’t match any RegEx, and then the program will tell the user such.

Project 3:

Due to the way I implemented project 2, the main goals of project 3 were already accounted for. So I took time to further optimize and improve the reusability of the code. I split various methods into relevant classes so that the code would be more readable and specific functionality could be isolated. I had a UI class that contained the user interface methods, and a Processor class that was used to analyze the user requests. The additional Main class then would create objects of the UI and Processor classes and call their methods as needed.

Project 4:

Due to the requirements of project 4, I had to remove a lot of the RegEx functionality and instead shifted to an intent map. User requests were compared to a list of words in an intent map using simpson coefficient, which would return a percentage of their overlap. If their overlap was greater than 0.7 then they would be added to a list. The user would then be asked which item in the list they were looking for when asking their question. This design model improved the functionality of my code a decent amount, but there were some gaps due to not including enough words in the intent map for each part. Due to this, I took some of the mapping from Ethan’s own intent map and added it to mine. His code was in Java, but I extracted the map by changing braces to brackets using ChatGPT.

Project 5:

For project 5, I adjusted my code from project 4 to track the session stats and append it to the end of the output file. The output file would be saved to a folder containing all of them. A new class, SessionStats would then take those files and update a CSV file to track all of them. It also contains methods that allow interaction between the user and chatbot, so the user can look for specific things such as a summary or a specific chat.

Project 6: (Integration)

All of the previously mentioned code was combined to create the final chatbot. As project 4 built on the previous projects, there wasn't much to change there other than implementing Tarun's project 1 and Ethan's intent map. As mentioned previously, those implementations weren't difficult and provided additional functionality. The integration with project 5 though was a bit difficult. It took a while to understand how to properly run a separate process in Python and not conflict with the current one. But the finished product properly calls project 5 from project 4 and opens as a sessions statistics menu.

Evaluation:

My chatbot had a precision of 85%, recall of 83%, for an F1 score of 84% across 20 requests. In terms of neatness though, it could be improved. I did not separate user input from chatbot output so the chat files look messy. Otherwise, the main functionality of the chatbot is decent.

Discussion:

Throughout this project I learned more about Python, class design, and chatbots. I understand more about how to make my code more reusable and how to reuse others'. Overall, I had a pleasant experience through this project and look forward to experiencing similar things in the future.

Conclusion:

The goal of the project was to create a chatbot which was achieved. The process was challenging but overall an enjoyable experience. I also learned more about how to make my own code more reusable and how to reuse others’ code.