Team RSJB

CS 410 Progress Report

1. Progress made thus far

Thus far, we have mostly been performing research and experimentation, as this is the first time that many of us have encountered the technologies to be used. Jeremy Bao has done some research into how BeautifulSoup is used and has figured out how to extract the text content from posts on StockTwits. He has also figured out how to extract the date from StockTwits posts, though the method does not seem to work on recent posts. His exploratory code can be found in "src/Try_Beautiful_Soup.ipynb". Sam Lee has developed a simple working search algorithm in the "src/search_algorithm" folder. He designed the search algorithm using the rank-bm25 package instead of metapy, because it seemed like a better fit within the scope of our project. Sam has also continued to research the best way to find the best F1-score and mAP from an improved search function. William Skedd has been learning about how Firebase is used, in order to find the most optimal way to store, manage, retrieve, and update data. His current plan is to retrieve the posts scraped by Jeremy and store them into Firestore, with each post stored as a document. This will give Sam access to the corpus of documents he needs to start testing his search algorithms. William has also begun to develop models for storing documents and caching query results. Ritik Kulkarni has been researching best practices for Flask development and UI design. Ritik has planned that there will only be a few front-end HTML pages needed, and how to communicate with Firestore DB when data querying and search functionalities are needed. UI design will be comparable to most current search engines in use to keep our site up to standard.

2. Remaining tasks

Jeremy Bao still needs to figure out how to store the obtained posts and implement the crawler program as a whole. Probably, each time it is run, it will obtain all posts up to the current day, not looking at any that have been previously crawled. Old posts do not have to be revisited, as editing posts is impossible in StockTwits. This will need an index to be recorded on the last visited post to make scraping easier. The process on how to store each post is still being refined. This will be decided by the team during the week as proposed by William Skedd. The options are to store versionable text documents into storage, to store each as an individual object, or both. Storing each post as an individual object may be needed to correctly link to a post on site for redirect purposes, but storing as a file in a format that can easily be read into Sam Lee's search function is the main priority. The current remaining tasks pertaining to the search algorithm are parameter tuning, creating example queries and judging relevance, as well as finding best F1-score and mAP. A cloud/lambda function needs to be deployed, which will hold Sam's search function, so that it can efficiently process search queries irrespective of the system. If a query is stored, then it will return an indexed result, else, it will run the search function, store query/result, and return result. There will also be a flag in Firestore that will inform the search algo cloud function on if new posts have been scraped and that the search algo should rerun on queries and store new results. Also, development of the Flask web app is dependent on all the above remaining tasks, especially proper management of database and cloud functions. Ritik will be designing an optimal UI layout to ensure that functionality occurs on the page once a search result is returned. This will be done locally until lambda/database is ready to be connected.

3. Any challenges/issues being faced

The current challenges that the team is facing is that we are not aware of the best way to deal with very rare words, such as usernames. While these may appear in just a tiny handful of posts, there are a massive number of them. Certain specific numbers or combinations may be even more unique, perhaps only appearing in a single post. Additionally, extracting the date from posts seems to be causing problems. While Jeremy was able to extract the date from old posts, more recent posts (that were made a few days ago) have the string “now” as their date, even though an actual date is visible on StockTwits for those posts. This may lead to an issue where giving the wrong or assumed date can skew the user's view or trust towards the project. Finally, deciding the most efficient way to store data and form a connection between all the project modules may prove difficult as the project has many components in place that are very dependent on each other.