Team MPDL Final Project Proposal: CS410 – Fall 2022
10/23/2022

Names, NetIDs, Captain
Team MPDL consists of the following members (NetID): Michaela Horn (mmhorn2), Peter Jefferson (peterwj2), Da'Mon Morris (dmorri25), and Lookman Olowo (lolowo2). The captain of the team is Peter Jefferson (peterwj2).

What is Our Topic?
Our team chose the 'Theme 5: Free Topics' category for the CS410 final project. Our free topic is to perform a sentiment analysis on news related trending topics that are scraped from Twitter's 'Trending' section and/or directly from news company's accounts (e.g., ABC, CNN, Fox, NY Times) to rank whether or not the posts are biased, subjective, or polarizing in the way they are written. The application we plan to build will offer a user-friendly GUI for the user to input a query for a news topic, select which news/media platforms to search on twitter, and then provide the ability to sort the results by bias/subjectivity.
An example of how a user would use the application would be to first, submit a news related topic query into the GUI, select/deselect which news sources they would like analyzed, and hit 'search'. Next, the application would call the Twitter API and search the selected news source's Twitter page(s) for the topic and return the text to the application's backend. With these results from each news source, we would then utilize the TextBlob Python library to perform a subjectivity and polarization analysis on the text. The TextBlob's 'sentiment' property will return a namedtuple of the polarity and subjectivity, respectively. Next, we would organize the results and display them back to the user in the GUI. The GUI will allow the user to order the results by subjectivity or polarity by high, low, or neutral (neutral meaning closest to 0 in this context). This enables the user to find the news organization(s) they want to subscribe to based on their subjectivity or polarization preferences.

Why is it Important or Interesting?
This is important and interesting because now, more than ever, news is presented with lots of bias, subjectivity, and polarization. It can be hard to figure out what is objectively true due to opinions impacting how some news organizations present news stories. Users can quickly find out which news organization to subscribe to based on bias results, or just find the least biased articles/Tweets on any given topic to get the most objective information on a given topic. It will also be interesting to study how the subjectivity and polarity differ by news organizations given a topic polarized across political party lines, and if the result will reveal that news sources are more liberal or conservative leaning.

What is our Planned Approach?
Our approach is to first create a GitHub project for our team members, create a base project and upload it to a main branch, and then create feature branches for each member's feature that they are working on.
Then, we need to configure a local REST API service using Python libraries. We then need to design the front-end GUI to allow a user to search a topic, select which news sources they want searched, and send the user-selection data to our backend service to search Twitter and return sentiment data.
Next, we need to make a shared Twitter account, sign up for the API service, and learn how to use it to query a news account and/or Twitter's 'Trending' topics. Once we're able to return those, we will then work on organizing the returned Twitter text by news source and then feed them into the TextBlob's 'sentiment' property to calculate the text's subjectivity and polarity. Once this is completed and stored locally for each returned news source's text, we will then

return to the GUI’s results page and allow the user to sort by subjectivity and polarity. We will also enable the option for a user to download the results to a CSV file to save the query results.

What Tools, Systems, and Datasets are Involved?
The tools we plan on using are Python for the backend, Typescript and HTML for the design of the user interface.
The systems we plan on using are a local web server to host our application locally by utilizing TextBlob Python library for the sentiment analysis, and Python libraries Flask for hosting a REST API, and Requests for interacting with external data sources. We also plan on using Angular and Typescript for the front-end. We will use GitHub as a system for our version controlling and collaboration.
The datasets involved will be the news source’s Twitter page posts related to a user-input query topic. Also, the TextBlob library vocabulary is utilized for sentiment analysis.

What is the Expected Outcome?
We expect the outcome of this application to be an easy-to-understand chart of the topic, its returned results from Twitter news accounts and their respective subjectivity and polarity scores (numerical values), a link to each of the returned Twitter results, and a downloadable CSV report of their query results. We will also include a sorting feature to the output so that the user can sort be high, neutral, or low subjectivity and polarity scores.

How do we Plan on Evaluating our Work?
We plan on evaluating our work by testing out the application with topics that are naturally high in subjectivity and polarization (e.g., topics like abortion rights, firearm laws, etc.). We will then manually read the returned Tweets/articles, rank them by our own thoughts of how subjective and polarizing they are, average our group’s ratings, and compare them to the application’s output. Another way we will evaluate our work is by comparing the results for the same topic by date. For example, if we ran the application for the query “healthcare” and got a subjectivity and polarity score of .4 and .3 respectively, we would download those results to a CSV, and then run that same query at a later date to see if the scores are similar. This will test the on-going accuracy of the application.

What Programming Language(s) are you Going to Use?
For the front-end/design portion of the application, we will be utilizing Typescript and HTML.
For the backend/functional programming, we will be utilizing Python because our team is comfortable with it, the Twitter API interfaces with it well, and the sentiment analysis library we wish to use, TextBlob, is a Python library.

Workload Justification/Estimation

Given our team has 4 members, we expect this application to take at least 80 hours (4 * 20 hours) to complete. See below how the hours are split up by task, team member(s), and time estimated to complete.

<u>Build Local Server (Flask REST API)</u>:

- Team Members: Lookman, Da'Mon
- Time Estimation per person: ~**10** hours as neither member has used Flask

<u>Build/Design Front-End (Angular Query Page)</u>:

- Team Members: Peter, Michaela
- Time Estimation per person: ~**10** hours as we are novice at front-end technologies.

<u>Establish to/from Communication with Twitter API & Make Account</u>:

- Team Members: Lookman, Da'Mon
- Time Estimation per person: ~**5** hours

<u>Learn how to interface with TextBlob's properties and format the results</u>:

- Team Members: Lookman, Da'Mon
- Time Estimation per person: ~**3**

<u>Build/Design results page with sorting dropdown, hyperlinked results, error handling, and exporting the CSV as a download</u>:

- Team Members: Peter, Michaela
- Time Estimation per person: ~**10** hours

<u>Testing, Validation, and Bug Fixes</u>:

- Team Members: All
- Time Estimation per person: ~**5** hours