Team MPDL Final Project Proposal: CS410 – Fall 2022

11/14/2022

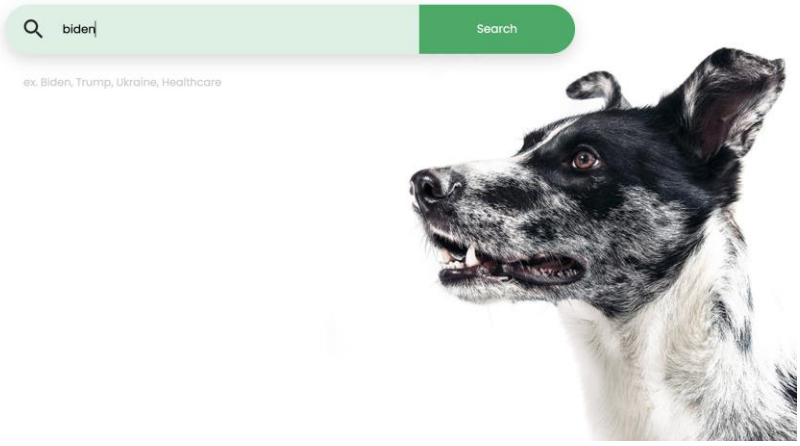
Names, NetIDs, Captain

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).

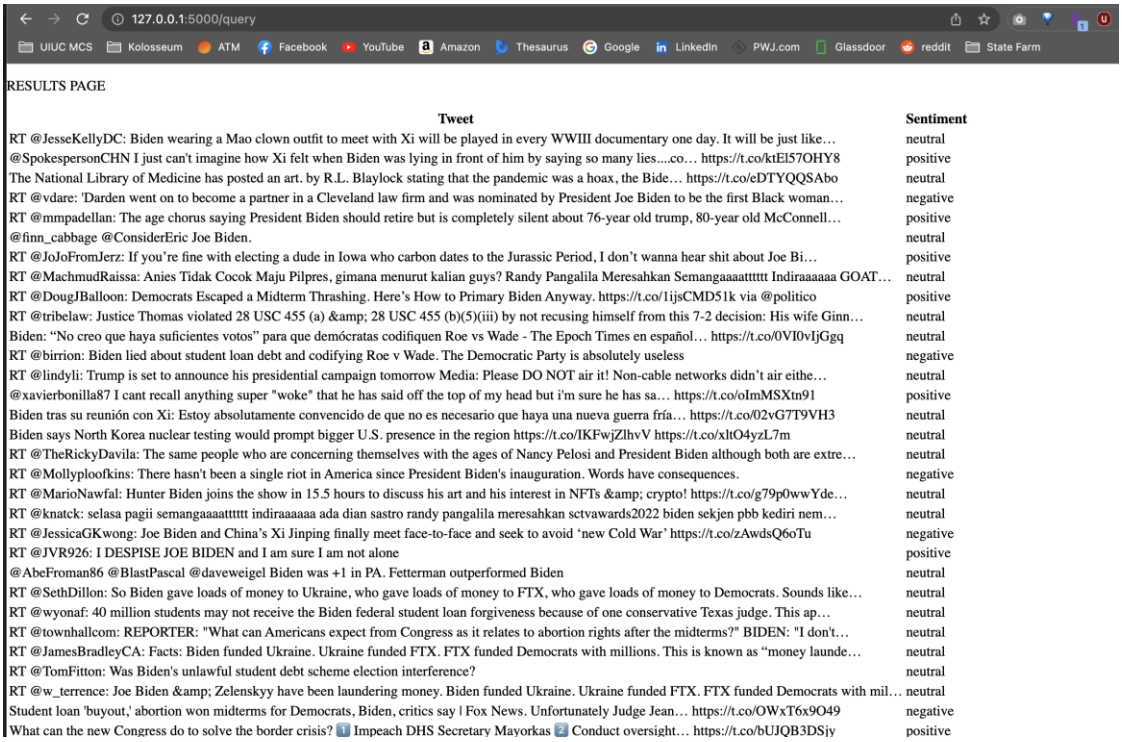
Which Tasks Have Been Completed?

Since our project proposal, our team has held a few meetings and accomplished some of the main tasks for the project. First, we created a Twitter account and set it up so we can utilize the API feature through it. We wrote a Python script to interface with the Twitter API that returns tweets for a given query.

We also implemented a Flask framework in our project and built out a simple search bar page as well as a results page.



We then linked the back-end and front-end so that when a user inputs a query and clicks "Search" it takes them to the results page and displays the results as well as the tweet's sentiment.



Lastly, we have been researching and redesigning portions of our projects in response to some of the feedback we received. We are adding more material from the course as a TA suggested. To do this we will not only be performing a sentiment analysis on the tweets, but also utilizing topic modeling to suggest related topics to search to the user once they search a few times. We have inserted a Draw.io diagram in the GitHub repository as well.

Which Tasks Are Pending?

We still have quite a few things to add on to our project. First, we need to find a way to save a user's queries so that we can suggest similar topics to them to query. Next, we need to implement the topic modeling algorithm into our back-end. We also need to write a couple of functions to clean, prepare, and return the queried tweets in a structured manner so that they do not have emojis, special characters, and are formatted as a 'document' in our returned results.

After, we also need to design the front-end to display the results cleaner, offer a filtering or ranking option, and make the tweets clickable. We will also need to include the options to filter a query by certain Twitter news pages if the user only wants to see results from certain news sources. This will most likely be accomplished via the use of HTML check boxes.

Lastly, we will also need to make the application more resilient to handle errors, null entries, and other bugs. We then will need to test the app by reading articles/tweets the are returned

for a query, deciding what our sentiment is on the article/tweet and comparing it to what the application returned.

Are you Facing any Challenges?

So far, we have only had a couple of challenges. The first being everyone's schedule lining up for the past couple of weeks; everyone has been very busy. We also still need to figure out how to implement the class algorithms into our application for our use case. We have been looking at past lectures and MPs to figure out how to best approach implementation of the topic modeling portion into our application.

In addition, we need to implement a way to save user's queries so that we can use them to suggest other related topics to search. We plan on using a local CSV file for this, but we may need to move to something more resilient like a database.

We are all also new to Flask, so we have been slowly coming up to speed on how to properly use the framework, and how to communicate between the back-end and front-end.