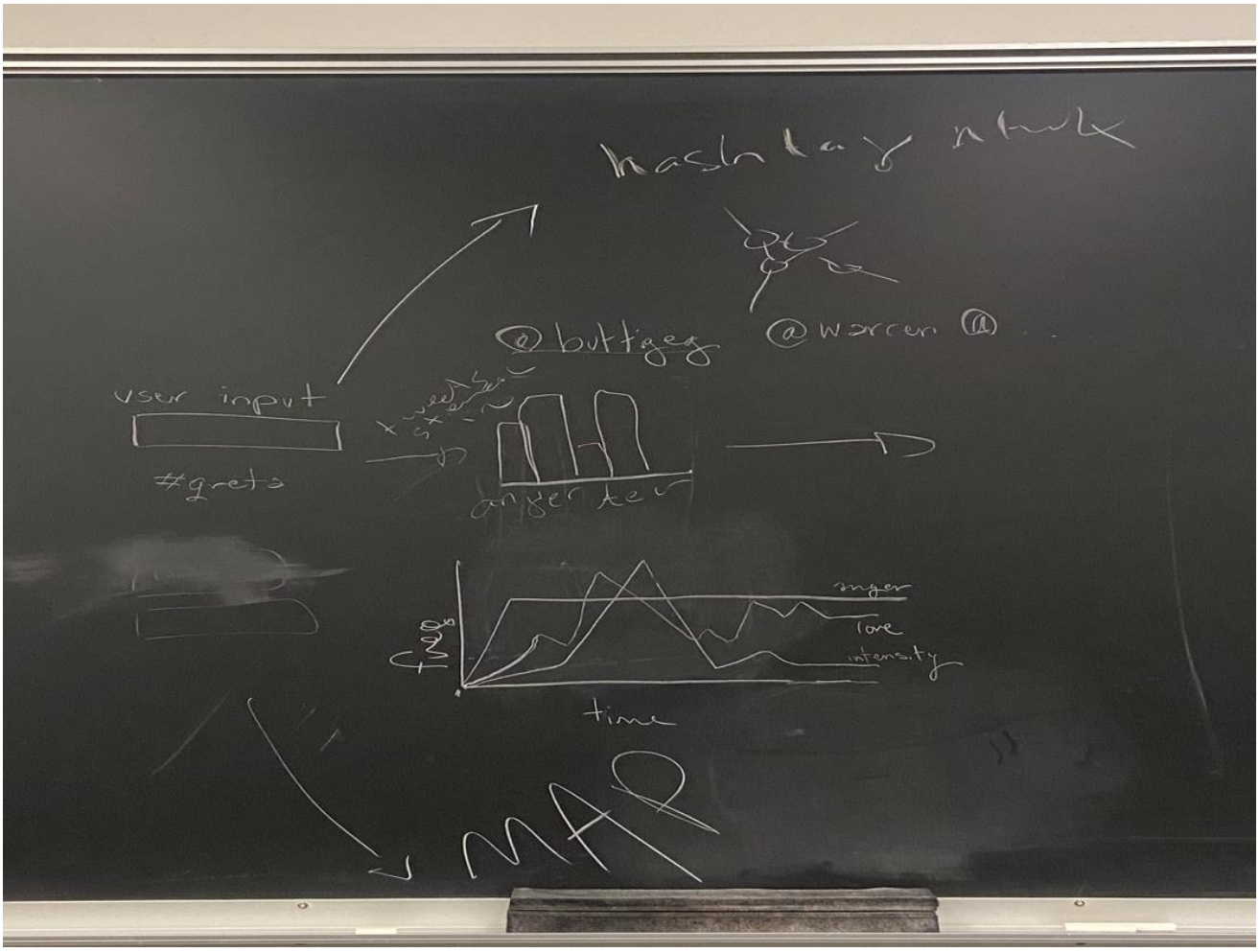
Data Visualization's Project Process Book

The 3 of us took Natural Language Processing last semester and were taking Social Network Analysis this semester as well, so it seemed natural to focus on a project that would combine elements of both disciplines. From the onset we also wanted to present something dynamic and "livestreamable", as opposed to simply displaying graphics on a static webpage.

You can find a screenshot of our first brainstorm session:



So we set off to write code using Twitter's API in Python, leveraging the existence of the StreamListener object to live-stream tweets after inputting a #hashtag. We built code to stream the tweets in, apply some basic pre-processing (removing stop words, funky punctuation, filtering for

language etc.) and initiall, classify them as being positive or negative based on a pre-trained machine learning classifier that we had built. The idea was then to show a running barchart and/or line plot of the ratio of positive/negative tweets as they came in.

We also wanted to create a network visualization of the incoming tweets, for instance creating ties between tweets with the same sentiment. Another idea we had was to leverage the geographic information contained in the tweets and plot them on a map. The infrastructure was initially planned to be in Python using a local webserver with Flask, but despite finding many similar repos on GitHub, it turned out that deploying multithreading with asynchronous support with an SQL server to store the tweets proved too challenging given the time constraints.

We took a step back and settled with a simpler and more doable approach. We replaced the sentiment ML model using dictionary matching with the NRC sentiment database, because training a multiclass classifier for 8 different emotions would have been too time consuming and would make the processing time too inefficient. We still kept the Twitter live-streaming idea, as well as the dynamic display of our results in forms of bar charts and running line plots.

We changed the display architecture to shiny dashboards, since they are simpler to use and we became familiar with them during the final weeks of the course. Working with threading and live updating within the shiny dashboard remained difficult, but we eventually figured it out.

After debugging our code in R, we figured that the best way for our message to come across, was by setting a time interval for our listener to rain in a given dataset (with the gathered tweets) and run our analysis using that information. The end result was an app which outputted one bar graph for the overall composition of thee tweets, a line graphs that showed how the sentiments had evolved through time and a map showing where the tweets we had listened to were being checked-in on. This last one, was subject to changes since people rarely check-in on twitter.

Gabriel Gilling
Cecilia Cabello
Lemon Lin Reimer