Using Twitter Data to Model Relationships Between Sentiment and Followers

Alex Taylor and Hannah Striebel

Project URL:
https://github.com/tayl1184/CSE482-Project

ABSTRACT

With the upcoming 2020 presidential election, social media platforms such as Twitter are a popular place to discuss political opinions. The goal for this project was to leverage Twitter data to better understand the current political climate. This project sought to collect and analyze Tweets to determine users' sentiment towards major media outlets, as well as discover and model any relationships between users and their sentiments towards the media outlets.

We collected data using the Tweepy python library, and then preprocessed the data to extract sentiment and follower data. For each media outlet, the resulting data set was used to train various classification models to attempt to predict a user's sentiment toward that media outlet based on which Twitter users they follow out of a predetermined set of users.

In the end, the accuracies of our predictive models were not concise enough to come to a conclusion. There may or may not still exist a relationship between user sentiment towards the media outlet and the users they follow. We were unable to achieve our goal of understanding the political climate through the data that we've collected.

1. INTRODUCTION

With the advent of social media, it has never been easier to share your opinions with strangers on the internet. We live in a time of an incredibly divisive and polarized political climate, full of strong opinions on both ends of the spectrum, loud voices, and incredibly antagonistic views of people with opposing opinions. Even the media outlet you choose to get your news from has political implications.

The current polarized and divisive political climate is intricate and complicated. Through this project, we sought to leverage Twitter data to better understand the current political climate. We first wondered if we could do so by analyzing sentiment towards a few main media outlets. Then we wondered if we could take it a step further - are there relationships between who a user follows and their sentiment towards a news outlet? For instance, would someone who follows *@realDonaldTrump* but not *@BarackObama* have a positive sentiment towards Fox News and a negative sentiment towards CNN?

Before discussing details of the project, it is important to understand what is meant by *sentiment analysis*, and how it will be used. From the Oxford dictionary[1], sentiment analysis is "the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude is positive, negative, or neutral." This will be useful to gauge the political climate of Twitter users by determining their opinion of specific media outlets.

The goal of this project is to gain a deeper understanding of the political climate by determining what relationships (if any) exist between whom a Twitter user follows and their sentiment towards specific media outlets.

We plan to achieve this goal by collecting Twitter data, preprocessing to extract sentiment and whom the user who tweeted follows, and then training and testing classification models using follower data as training attributes and sentiment as the class attribute. If relationships exist between sentiment and users whom they follow, our classification models will have a high training and testing accuracy, and we can use these models to gain political insight. However, it is also possible that no such relationships exist, in which case our classification models will have low accuracy.

The data set we collected for this project consists of Twitter data collected over the course of a month using the Tweepy python library. We chose our search keywords to be the Twitter screen names of five major media outlets: "CNN", "FoxNews", "MSNBC", "NPR", and "cspan". After collecting our data, we began preprocessing by analyzing the sentiment of the text of each Tweet. Then, due to challenges (discussed below), we took a sample of our collected data set, and performed further preprocessing to format the data to be used for classification modeling (Table 4). To perform the classification task the data is divided into 30% for testing and the remaining 70% to train the model.

The main challenge we encountered was that Twitter rate limits calls to its API. This means you can only request the API to return information to you a certain number of times during a given period of time. This posed a problem during the late stages of preprocessing and analysis.

To format the data correctly to be used for classification, we needed a list of users that are going to be the basis for the relationships we are looking for. Originally, we wanted this to be determined from the dataset we collected - the users who were followed most by the users whose Tweets we collected. We planned to do this using the Misra Gries algorithm where the input data stream was, for each tweet, querying the API to get the list of users that that user follows. However, this API query is rate limited to 15 requests per 15 minutes, and we had collected tweets

from over 20,000 users. At that rate, it would take over 13 days to perform this analysis.

As a result, in order to perform our preprocessing in a realistic amount of time, we needed to decrease the number of "top followed users" we were testing against, as well as decrease the number of Tweets we were processing. As an alternative to extracting the top followed users from our dataset, we used the top followed users on all of Twitter. In order to narrow this list down for relevance, we choose the top 20 Twitter accounts that were related to news and/or politics[2], including the news outlets we chose as our keywords. (This list can be found in top_followed.csv on our github repository, or below in Table 3.) Then, in order to decrease the number of Tweets we had to process, we took a random sample from the data we had collected. Even with these steps, it took over two days to run the preprocessing found at classification_preprocessing.ipynb

Due to the low accuracy of our prediction models we are unable to say for certain that a relationship does or does not exist between the user's sentiment towards media and the Twitter accounts they follow. This being the case, we did not meet our goal of understanding the political climate via Twitter.

CSE881-2015, Month 1–2, 2004, City, State, Country.

2. DATA

The data set we collected for this project consists of Twitter data collected over the course of a month using the Tweepy python library. We chose our search keywords to be the Twitter screen names of five major media outlets: "CNN", "FoxNews", "MSNBC", "NPR", and "cspan". Using the Tweepy search API, we collected 100 tweets for each keyword every day from February 15th through March 10th. We collected additional data using the Tweepy streaming API on March 3 (Super Tuesday).The code used to collect this data can be found on our github repository under data_collection.ipynb.

Total number of Tweets	**23,980**	**378 MB**
Collected with Search API	1,150	58 MB
Collected with Streaming API	22,830	320 MB
Number of Tweets for CNN	9,725	146 MB
Number of Tweets for cspan	429	14 MB
Number of Tweets for FoxNews	4052	65.5 MB
Number of Tweets for MSNBC	8379	126 MB
Number of Tweets for NPR	1395	26.5 MB

Table 1: Summary statistics of the raw data from Twitter

The data collected was stored in a separate .json file by keyword and by day. These .json files contained the entire Tweet object returned by the Twitter API for each Tweet that was collected. A number of attributes were used in preprocessing (summarized in the Table 2 below), but the most relevant attributes are *screen_name* and *text*, as *text* will be used to extract sentiment, and *screen_name* will be used to determine who that user follows.

Attribute name	**Type**	**Description**
screen_name	Nominal	Name of user who Tweeted
text	Nominal	Contents of the Tweet
lang	Nominal	Language of the Tweet
retweet_count	Ratio	Number of times this Tweet was retweeted
friends_count	Ratio	Number of people the user who Tweeted follows
followers_count	Ratio	Number of people who follow the user who Tweeted

Table 2: Attributes of the data acquired from Twitter.

The first preprocessing steps (found on our github repository in sentiment_preprocessing.ipynb) was to analyze the text of each Tweet to extract the sentiment. We implemented this by following a tutorial on geeksforgeeks.org[3]. We used a python library called TextBlob, which is a high level library built on top of the Natural Language Toolkit (NLTK) library. When creating a TextBlob object from text, sentiment analysis is performed, and stored in the object as a property '.sentiment.polarity' that is a floating point value within the range [-1.0, 1.0]. We used this property to classify each tweet as 'positive' (polarity > 0), 'negative' (polarity < 0), or 'neutral' (polarity = 0).

At this point, we threw out Tweets that weren't in English. The sentiment analysis library we used is only trained to recognize sentiment in English words, so including non-English Tweets would have falsely increased the percentage of Tweets analyzed to be 'neutral'. In order to look only at unique Tweets, we also threw out Tweets where the text was a direct retweet of another Tweet.

During sentiment preprocessing, the data collected for each of the five media outlets was consolidated into a single file per media outlet. For each Tweet, an additional attribute called "sentiment" was added to the json object with the results from the sentiment analysis on the text of that Tweet.

The next step in preprocessing was to take a sample of the collected Tweets, and then go through the sampled Tweets and obtain the "top followed" data in order to format the data for training a classification model. (This can be found in classification_preprocessing.ipynb.) We sampled the data with a random sample, and at this point we also threw out Tweets where the user who Tweeted had zero friends and zero followers.

The format we used for classification is as seen in Table 4. Each row represents a Tweet. The class attribute is the analyzed sentiment of that Tweet. Each of the 20 predictor attributes are a

binary 1 or 0 whether or not the user who wrote the Tweet follows that use.

This was obtained by first looping through the unique users whose Tweets we collected, and testing if they follow each of the top followed users (found in top_followed.csv, and in Table 3 below), and writing their username along with the follower data as a 20 character bit string to a .csv file. Then, leveraging this data, we looped through each Tweet in our sample, and combined the "sentiment" attribute (that we extracted earlier in preprocessing) with the bit string for the user who wrote that Tweet.

	rank	screen_name	millions of followers
0	1	BarackObama	115
1	9	realDonaldTrump	76
2	16	cnnbrk	57
3	22	CNN	47
4	24	nytimes	46
5	30	BBCBreaking	42
6	39	NASA	36
7	47	PMOIndia	33
8	61	POTUS	29
9	69	BBCWorld	27
10	70	HillaryClinton	27
11	84	TheEconomist	24
12	92	Reuters	21
13	94	WhiteHouse	21
14	115	FoxNews	18
15	131	WSJ	17
16	140	TIME	16
17	499	NPR	8
18	1928	MSNBC	3
19	3544	cspan	2

Table 3: Top 20 followed political and news accounts on Twitter

	sentiment	BarackObama	realDonaldTrump	cnnbrk	CNN	...
0	neutral	1	0	0	0	...
1	neutral	1	0	0	1	...
2	negative	0	1	0	0	...
3	neutral	0	0	0	0	...
4	neutral	0	1	0	0	...
...	...	...	...	...	...	...

Table 4: Sample from our cspan preprocessed DataFrame object

For each of the five media outlets, our final, preprocessed dataset consists of one class attribute - "sentiment," which is positive, negative, or neutral - and 20 predictor attributes - whether or not the user who Tweeted follows each of the top 20 news/political accounts on Twitter. Our final dataset is smaller than we would have liked it to be; however, we used as large a sample as possible, given that Twitter rate limits queries to its API and we were limited by time.

Total processed Tweets	**2,046**	**102 KB**
Number of Tweets for CNN	751	37 KB
Number of Tweets for cspan	87	5 KB
Number of Tweets for FoxNews	373	19 KB
Number of Tweets for MSNBC	678	33 KB
Number of Tweets for NPR	157	8 KB

Table 5: Summary statistics of the processed data from Twitter

3. METHODOLOGY

We first collected data from February 15 through March 10th. Then, we performed a first round of preprocessing to extract sentiment from the Tweets. We then determined which top followed users should be used as our predictor attributes. The second round of preprocessing processed the users who wrote the Tweets we collected to determine whom of the predictor attribute users they follow. We then combined this data with the sentiment data extracted earlier to get our final dataset.

We applied several different classification methods throughout the analysis of our data to ensure our results were legitimate. We implemented a decision tree classifier, a logistic regression classifier and a neural network classifier.

Our data is divided into 30% testing and the remaining 70% for training using the train_test_split function from the sklearn library. Both the decision tree and logistic regression models featured cross validation to select the model's hyperparameters. This selection was done independently for each news outlet data set. We did not use any method of parameter tuning for the neural network.

Our project code is organized in the following files:

- data_collection.ipynb: this is the Jupyter notebook file to collect data from Twitter using the Tweepy library.
- sentiment_preprocessing.ipynb: this is the Jupyter notebook file to perform sentiment analysis on the collected Tweets.
- classification_preprocessing.ipynb: this is the Jupyter notebook file to finalize preprocessing by processing the users and determine who they follow.
- Modeling.ipynb: this is the Jupyter notebook file to perform the classification task of the project and compare accuracies between models.

We used the Tweepy python library to collect our data from Twitter. In order to complete our sentiment analysis on the text of the tweets we used TextBlob, which is a high level library built on top of the Natural Language Toolkit library. The classification models, as well as the cross validation and accuracy score functions are from various sklearn libraries. In addition, and although they are not necessary for the replication of this experiment, MatPlotLib, NumPy and Pandas were all used for organizational purposes.

4. EXPERIMENTAL EVALUATION

4.1 Experimental Setup

Computations and code were performed and written on a Windows computer with no additional hardware. We used a portion of our dataset reserved for testing as our baseline. If a model trained on the training portion of our dataset performed well on the never-before-seen testing portion of the dataset, then we would know that a relationship had been found. This performance was evaluated by test accuracy through the sklearn library.

4.2 Experimental Results

Our experiment initially started with the data being passed through a decision tree model to tune the maximum depth parameter for each news outlet's dataset. For each iteration we performed 10-fold cross validation to determine the score at the given depth. From there we chose the highest scoring depth to be the optimal hyperparameter for that dataset's decision tree model. However, the accuracy yielded in this way was too low to express any relationship in our data, the average accuracy across all decision trees being 0.454 as seen below in Table 5. To look even further for a possible relationship we passed the same datasets through logistic regression models and neural networks. Once again using multiple iterations to tune our hyperparameters. All of which returned low accuracies.

Model Type	Decision Tree	Logistic Regression	Neural Network
Avg. Accuracy	0.4541	0.4589	0.4369

Table 5: Average accuracies across all datasets for a given model

We then merged our datasets in an attempt to see if a relationship exists between the Twitter accounts, in regards to news/politics, and an overall sentiment. After following the same process as with the subsets and comparing accuracies between models, our best result was found to be an accuracy score of 0.4723 using logistic regression. When looking at the coefficients created by the model, we see that none of the features provide much insight into the classification of the Tweet's sentiment. The two largest weights in terms of absolute value are -0.829 and 0.427 which allude to BBCBreaking and PMOIndia as being the most prominent predictors of determining a user's sentiment. After receiving similar results across all models and datasets we came to the conclusion that a relationship between the news/political accounts a twitter user follows and their sentiment towards media outlets cannot be determined with the data that we've collected.

One potential reason we were unable to find a relationship is due to the sparsity of our data. For example, out of the roughly 2000 processed tweets, only 37 people followed PMOIndia. Additionally, certain users did not follow any of the predictor attribute users. Although this is valid data, it may have made it more difficult to accurately adjust the weights. A potential solution to this problem would be to drop the rows that do not follow any of the predictor attributes and retraining the models.

Another possible approach to alleviate this problem would be to analyze and reassess which users should be included as predictor attributes. Decreasing the number of features may increase the change of uncovering relationships. We were unable to extract this list from our dataset as we originally intended, which most likely contributed to the sparsity of our resulting dataset.

5. CONCLUSIONS

As social media's presence within our culture grows more prominent, so will the need to understand its effects on society. While our work was inconclusive in finding relationships in the data we collected, it doesn't necessarily rule out the possibility that one exists.

When designing future projects, researchers could improve upon our results in a number of ways. It would be beneficial to do further testing and analysis on the number and content of features included for training. For example, excluding sparse attributes, or records where the user does not follow any of the predictor attribute users. In addition, it may be useful to look into other options for performing sentiment analysis in order to validate the sentiment results returned. One other possible area for future study would be to analyze the results from excluding 'neutral' Tweets.

6. REFERENCES

[1] "Sentiment Analysis: Meaning of Sentiment Analysis by Lexico." *Lexico Dictionaries | English*, Lexico Dictionaries, www.lexico.com/definition/sentiment_analysis.

[2] "Top 500 Twitter Users by Followers." *Social Blade*, Social Blade LLC, socialblade.com/twitter/top/500/followers.

[3] Kumar, Nikhil. "Twitter Sentiment Analysis Using Python." *GeeksforGeeks*, 7 Feb. 2018, www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/.