Natural Language Analysis of Twitter for the 2020 United States Presidential Election

Collin J. Kovacs
Mathematics and Data Science
Eberly College of Science
Pennsylvania State University
University Park, PA 16801
Email: cjk5642@psu.edu

Thomas Y. Zhao
Computational Data Science
College of Engineering
Pennsylvania State University
University Park, PA 16801
Email: tyz5069@psu.edu

***Abstract*—Over the past decade, Twitter has become a key area for (Western) political discourse via its system of Tweets and followers. Tweets are composed of unstructured natural language which beckons the use of natural language processing (NLP) tools to analyze it. A great deal of latent information about the world, trends, and underlying demographics of Twitter can be accessed through these Tweets with varying effectiveness depending on the NLP technique. Information from Tweets is particularly valuable with regard to politics, which data scientists have been trying to crack since the early 2010s. The past few years have seen numerous breakthroughs in NLP, offering novel and effective techniques whose potential in extracting useful information from Twitter have yet to be fully explored. We apply modern NLP techniques and models to the task of gaining insight regarding the Twitter landscape in the week leading up to the 2020 US Presidential Elections. Through sentiment classification, tone, and topic modeling, our results suggest key differences on Twitter between the 2016 and 2020 election years, such as more negative sentiment on average in the 2016 election compared with 2020. We compare two techniques for sentiment analysis, a Naive Bayes classifier and a state-of-the-art classifier from the flairNLP Python package. Our results indicate that improvements in NLP and machine learning will continue to enhance our ability to extract meaningful information from natural language ”data mines” such as Twitter.**

I. Introduction

Twitter has been extensively studied as a data source that is reasonably representative of the broader populous, in addition to its geolocational data. According to research done in trying to predict heart disease, Twitter data has been shown to have equal or better predictive power compared to traditional demographic variables such as age and weight [Eic+15]. Researcher have also attempted electoral prediction [Gay13], and prediction of geolocation itself [HCB14], though with varying levels of success. We are interested in the problem of understanding the 2020 US Presidential Election. However, as this project is only a brief exploration of the topic, we will not be attempting any meaningful predictions. Still, we are aware that Twitter offers great predictive potential for subjects even as complex as a US Election.

Elections have been studied through Twitter for as long as Twitter has existed. Early approaches used simple metrics such as tweet volume and simple sentiment models to analyze an election[Tum+10]. More complicated metrics have also been explored, such as proportion of mentions [SCR12] and time series features [Bor+12]. While all these features have been found to be reasonably correlated with election results/party performance, they fail to make full potential of the heart of Twitter data: the Tweet. Tweets are the main messages being sent out by users (with a 140-character limit imposed by Twitter). The best way to make use of these messages is through natural language processing. This project solely focuses on exploration of NLP as it applies to Tweets, but we also believe NLP can provide better features that may improve our ability to predict elections, or perhaps supplement traditional polls.

II. Literature Review

Social media has revolutionized communication in the ”public sphere” and the way opinions are shared and received. The Twitter microblogging service in particular has dramatically shifted how society conducts political discourse, particularly in the United States. There is much to be gained from to analyzing, quantifying, and interpreting the political theatre that is Twitter. Understanding politics on Twitter is therefore a significant stepping stone to understanding the US political landscape as a whole. Twitter data has long been utilized and researched for problems ranging from prediction, classification, event tracking and more. As such, researchers have approached the relationship between Twitter and politics from a multitude of angles.

One common application of Twitter data is sentiment prediction. Previous studies have applied regular sentiment prediction (positive vs negative) on tweets from political candidates regarding various topics [VP20]. A second scenario is user classification, where researchers have aimed at predicting political leaning (conservative vs liberal in the US) through text [Rao+10], network connections [ALR12], or some combination [PP11]. More recently, researchers have taken a more generalized approach to predicting political ideology not limited to the conservative-liberal spectrum [Pre+17]. Another major research topic using Twitter data is identifying real world events [MP08] and predicting their spread [Che+14].

Sentiment analysis in relation to governance has already been applied in evaluating trust-levels of citizens [Cal+15] and predicting elections [Tum+10]. Researchers have proposed a measure called ”relative support” which seeks to quantify the relative strengths of political parties, and this parameter was utilized to analyze the 2011 Spanish Presidential Elections [Bor+12]. Another group also applied this technique to Italian elections, with reasonably strong correlations [Cal+14]. The method used the slopes of the time series of accumulated tweets mentioning each political party to measure their support on Twitter. The study claimed that user activity on Twitter correlated with the election results.

We seek to utilize existing NLP tools to gain insights into the Twitter landscape of the 2020 Election. A previous study of the 2016 Election used the freely available SentiStrength tool, which researchers used to explore sentiment relations between candidates vs citizens and citizens vs trending topics [Yaq+17]. The study found that Twitter does indeed accurately reflect the public opinion and important topics of concern regarding the elections, and that Trump's sentiment on Twitter was overall more positive than Hillary's, which was reflected in the election result. In our work we use more recent sentiment classification techniques, such as attention-based classifiers with contextualized word embeddings, which has been shown to be even more accurate at extracting sentiment information from natural language [ABV18].

Previous Twitter-related studies have focused on classification tasks (e.g. sentiment classification), and in doing so, tested various methods of word embedding [YMO18; PP11; Rao+10; Pre+17]. One notable finding was that researchers observed better classification performance when using word embeddings trained with a corpora more closely aligned with the relevant domain [YMO18]. More advanced word embeddings based on transformers have been found to make the best performing classifiers thus far, which our work utilizes [ABV18].

III. Data Set

We use both current and historical Twitter data, from the years 2020 and 2016 respectively, for a comparative analysis of the two seminal elections. All our data ultimately originates from Twitter, although we employ two different methods for obtaining the data.

A. Historical Data

For historical data from the 2016 US Election, we source data collected by Harvard Dataverse [LWK16].

This dataset contains approximately 280 million tweets relating to the 2016 US presidential election. These tweet ids are broken up into 12 collections. The collections are:

- Candidates and key election hashtags (Twitter filter): election-filter[1-6].txt
- Democratic candidates (Twitter user timeline): democratic-candidate-timelines.txt
- Democratic Convention (Twitter filter): democratic-convention-filter.txt
- Democratic Party (Twitter user timeline): democratic-party-timelines.txt
- Election Day (Twitter filter): election-day.txt
- First presidential debate (Twitter filter): first-debate.txt
- GOP Convention (Twitter filter): republican-convention-filter.txt
- Republican candidates (Twitter user timeline): republican-candidate-timelines.txt
- Republican Party (Twitter user timeline): republican-party-timelines.txt
- Second presidential debate (Twitter filter): second-debate.txt
- Third presidential debate (Twitter filter): third-debate.txt
- Vice Presidential debate (Twitter filter): vp-debate.txt

Within the context of the Twitter API used to gather these tweets, ”Twitter filter” refers to keyword and hashtag filters while ”Twitter user timeline” refers to pulling specific users' tweets.

We filtered for only non-retweets, since we are primarily concerned about the contents of each tweet. This was done by creating a new dataset on TweetSets and filtering for no retweets. The filtered dataset for the “Candidates and key election hashtags” can be found here: https://tweetsets.library.gwu.edu/dataset/a7bde71a. The filtered dataset for everything else can be found here: http://tweetsets.library.gwu.edu/dataset/324e94e8. After filtering, we are left with 94 million original tweets.

The historical data only provides tweet ids, so we must retrieve the text and user data using Twitter's API. The twarc library offers a command line tool and Python libraries to do this, so we used twarc's hydrate command to retrieve the complete tweet data.

However, one limiting factor is that Twitter limits queries to 900 requests of 100 tweet ids per 15 minute window per set of user credentials. This works out to 8,640,000 tweets per day, so hydrating the “candidates and key election hashtags” dataset (84 million tweets) will take 10 days, or 5 days if we use two separate user credentials. This is why we split the “candidates and key election hashtags” dataset from the rest of the data in case we do not have the time to analyze it all.

B. Current Data

To obtain our current data, we used a Twitter API called Tweepy. We then created Python scripts to authorize, obtain, preprocess, and analyze the data obtained from the Tweepy API.

1) Credential: In this script, we input our develop credentials such as: access token, access security token, consumer key, and consumer security key. From these credentials, we save them into a JSON file and store it in our Data folder.

2) Authorization and Obtainment: Here, with a series of functions, we authorize and obtain the required data. Given a function that reads in my credentials then authorizes them through the Tweepy API, we then, given a set of parameters, scrape tweets from Twitter which appends rows to the dataframe given the following parameters:

- Username of User
- Description of User
- Location of User

- Number of People the User is Following
- Number of User's Followers
- User's Total Number of Tweets
- User Creation Date
- Tweet Creation Date
- Retweet Count
- Hashtags in Tweet
- Tweet

We extracted tweets from the date range of October 25 to November 2. We use different keywords to search in hashtags like $election$, $trump$, and $biden$. Along with this, we ignored retweets and we removed tweets that did not provide a location when their tweet was posted. This gives us ample information to understand twitter users and the tweets they post.
During the scraping period of the day November 2nd, we ran into many bugs in trying to collect data using Tweepy. We suspect that this was due to a new massive inflow of Twitter users trying to scrape tweets just like us. This led to less API calls and an increase in errors through extraction which, in turn, led to less data collected on that day.

3) Preprocessing: In this script, we implement different functions to clean the text of each Tweet in our data set. One large task was the actual cleaning of the data. This consisted of the removal of punctuation, contractions, and emojis. The toughest part of this process was the removal of emojis since the $emoji$ library did not remove every emoji. Here, a list of Unicode statements were needed for every case.
The next largest task was the extraction of locations. The location data had no definitive pattern. Here, we extracted each location to make a standardized location which was the state name. For example, a location name of "Huntsville, AL" would be standardized to "Alabama" and "Wyoming, USA" would be standardized to "Wyoming". After the cleaning process was done, our data reduced from 233,618 rows to 89,644 rows of cleaned tweets and standardized location.

IV. Exploratory Data Analysis

To get a better picture of our current data, we perform some exploratory data analysis. Overall, we find that our data is heavily skewed toward more populous US states, such as California and Florida.

1) Box plots: One of the analyses we performed was obtaining the summary statistics of the number of words within each tweet, grouped by each states. Figure **??** plots a box plot of that information per state. This gives us a good idea of how long a general user's Tweet may be, and whether there are any outliers. This also provides insight into understanding each keywords efficiencies and how influential one state may be within our actual data. In this example, there are multiple box plots in Figure 1 that look at the word length for tweets. Even though nothing truly stands out from the graph, this helps give a shape of reference if in the future we need to sample under a distribution.

2) Correlogram: A correlogram is a pairwise matrix that identifies the correlation between every continuous variable or feature in a data set. In this context, we find that there are two correlations: tweet words and tweet letters, obviously, and retweet count and favorite count. In Figure 1, the first correlation is obvious, but the second makes sense as well since a person that retweets a Tweet will most likely favorite as well.

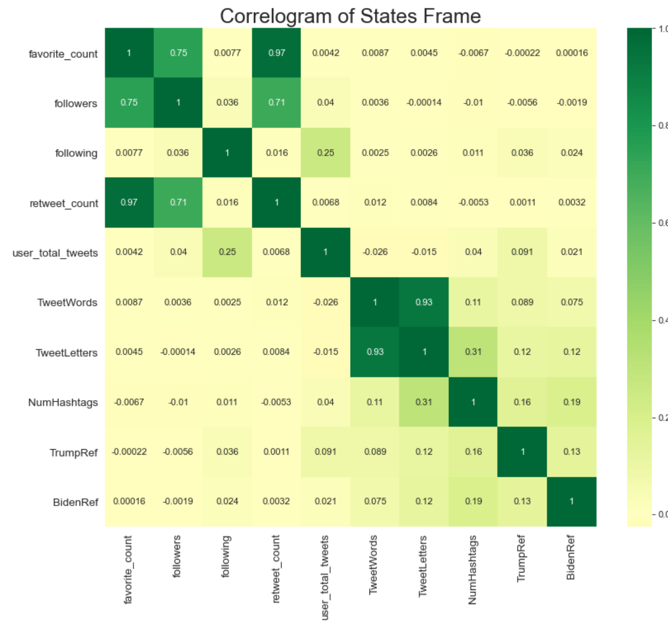


Figure 1: Correlogram of All Continuous Factors

3) Density Plot: A density plot consists of two parts, the histogram of the data and a density curve overlayed. We use it to understand the distribution of states in our data by the number of times each State appears. Like the other plots, we use this to understand the biases and skewness of our data. Figure 2 shows that our data is skewed right by a strong amount, meaning a couple of states provide a majority of tweets. It also shows that we have some strong outliers in our data. This is a characteristic that we would take into account in a future work for election prediction.

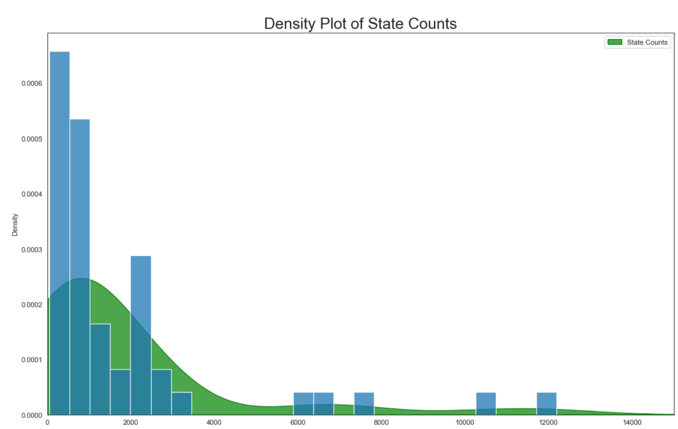


Figure 2: Density Plot and Histogram for Counts of States

4) Tree Map: A tree map is a diagram that takes categorical and continuous variables and maps it to a rectangle and sizes it by the number in the continuous variable. In our context, we look to understand the number of references of Trump and Biden by State. By references, we mean the number of times Trump was used in a tweet and summing the number of times that happens by State. In Figure 3, we can see that many ”blue” or Democratic states have really high counts. This can show bias towards our data since more Democratic or ”urban” people are more likely to use Twitter.

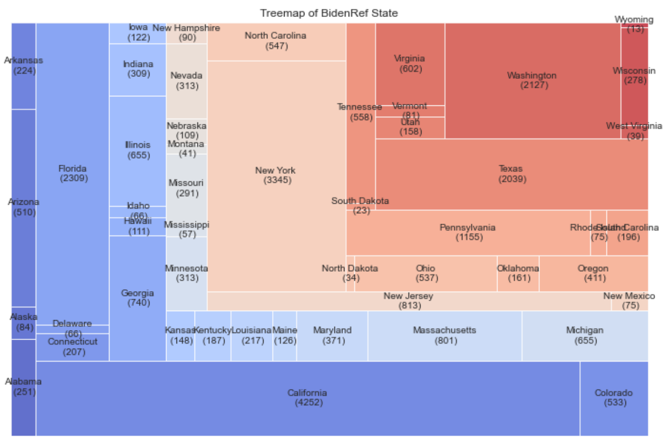


Figure 3: Tree Map of the Number of Times that Biden was Referenced by State

V. Methodology and Results

In this section we present the structure and flow of Tweet data through our various methods of analysis, as well as the results. Figure 4 shows our two main data sources, in addition to the NLP tools used. Note that all of our analysis was done in a Python 3 environment, using both Python scripts and notebooks. The Google Drive containing most of our Python notebooks is hyperlinked to this sentence.

A. Accessing the Data

We converted all the Twitter data we collected into JSON Lines format (JSONL), which makes accessing all our data simple and standardized. JSONL is JSON except each line in the file is an entire JSON object. This allows for simple line-by-line reading of JSON data, which can be conveniently performed in Python with a single generator method.

B. Data Preprocessing

Cleaning the data is a necessary first step for the parts of our methodology that did not utilize third party APIs. For Tweets, this involves converting all words in the tweets to lower case, removing numbers, hashtags, mentions, punctuation, and special characters. The text is then tokenized, or split into individual words. In the same process, we also add labels for parts of speech (PoS) to each word, which is needed by WordNet for lemmatizing. We lemmatize the Tweet text using the WordNet stemmer interface (WordNetLemmatizer), which is included in the Natural Language Processing Toolkit (NLTK) library for Python. Lemmatization refers to reducing the different forms of the same words to the same root word (e.g. the words “walking” and “walked” are transformed to “walk”). Next, we remove the stop words from the text. Stop words are words that have no informative-value which introduce noise in the data, such as “the” and “we” and other common words. Most NLP libraries such as spacy and nltk include a set of stopwords which we will use. However, we do not include negation words in the stop word set. Negations such as “not” or “no” contribute to the positive or negative sentiment of the tweets when using Naive Bayes for example.

C. Sentiment Analysis

One of our primary goals is to quantify Twitter sentiment toward each political party, which calls for some sort of sentiment classifier. Early work on this topic traditionally used either manually annotated sentiment models or Naive Bayes Classifiers [CM11]. For this project we chose two methods for sentiment classification, a Naive Bayes Classifier and a modern BERT-based neural network classifier. The math behind Naive Bayes is not complicated, but suffice to say, the Naive Bayes approach for NLP does not take into account a word’s context. We implemented the Naive Bayes classifier for our tweet analysis using the Python library TextBlob. TextBlob contains a ”NaiveBayesAnalyzer” module, trained on a corpus of movie reviews. Although the trained domain does not exactly match ours (movie reviews tend to be longer than tweets), the two models we test both trained on movie reviews, so we believe the comparison should be reasonable.

Recent breakthroughs in NLP have found novel ways to capture a word’s context, improving the performance of sentiment classifiers (among other uses). The keystone technology behind this achievement is a language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers [Dev+18]. Researchers building upon BERT have incorporated recurrent neural networks into new word representations, called Contextual String Embeddings [ABV18]. Thanks to the Python library called Flair, which those researchers provided, we are able to implement these cutting edge NLP methods with relative ease in our Python environment. Like TextBlob, Flair comes with a sentiment analysis module which is also trained on a corpus of movie reviews. Unlike TextBlob, the Flair classifier uses an Transformer-based architecture, in addition to Contextual Word Embeddings.

1) Sentiment Towards Candidates: We used sentiment analysis explored the differences in Twitter sentiment with regard to each political party. We perform this analysis on per-candidate basis, since due to API constraints, our current data only consists of tweets that mention one of the candidates’ names. We also grouped our data by states for a more granular and relevant analysis of the election. The primary metric we based our analysis on was the proportion of positive tweets (PPT) in each state, calculated as such:

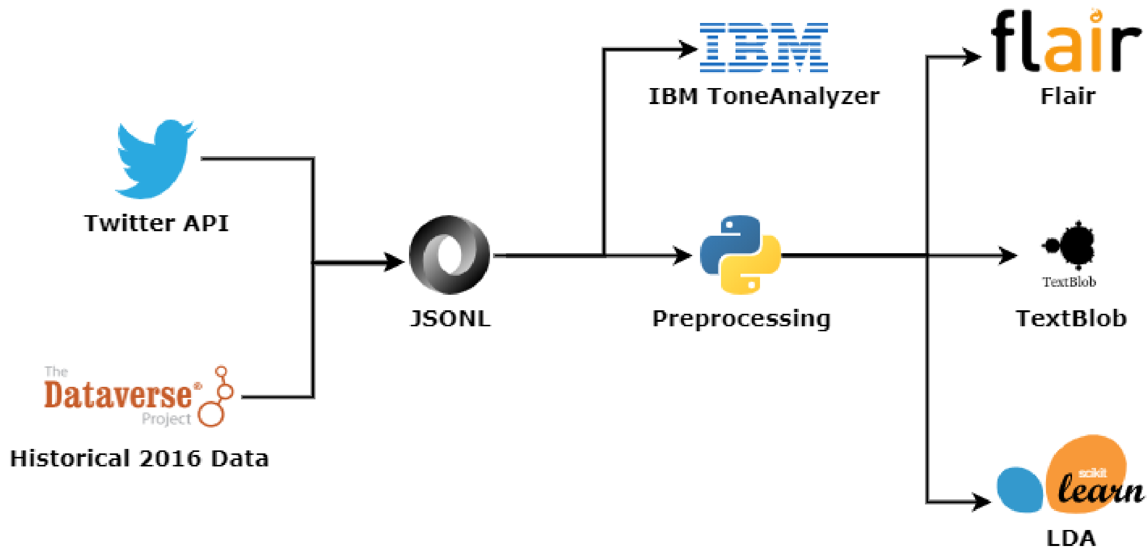


Figure 4: Overview of our data collection and processing pipeline.

$$\frac{\text{\# of positive tweets per state}}{\text{\# of total tweets per state}} \tag{1}$$

Figure 5 depicts the results of our sentiment analysis for 2020 (using Flair), after filtering for tweets that exclusively mention Trump or Biden respectively. We chose to ignore tweets that mention both parties' candidates since it is unclear to whom the sentiment is being directed (perhaps future NLP libaries will be able to analyze sentiment subject-by-subject). Also note that the color scale of each heatmap is relative to itself, it is only the numbers that should be compared between each figure. Figure 6 depicts the same analysis but for 2016.

Our results show that tweets mentioning Biden were generally more positive across all states, ranging from 26% positive (Utah) to 63% positive (Vermont). The 2020 tweets mentioning Trump were generally more negative, ranging from 12% (Delaware) to 38% (North Dakota). Interestingly, the results for 2016 were much different. As Figure 6 shows, tweets mentioning Hillary were overall more negative than Trump. PPT for Trump in 2016 ranged from 18% (Vermont) to 31% (Kentucky) whereas PPT for Hillary ranged from 15% (Virginia) to 36% (Delaware). Most notable is the overall PPT for each candidate over all of Twitter, where 22.7% of Trump-mentioning tweets were positive compared with just 19.1% for Hillary-mentioning tweets.

Although these metrics say little about actual voters, it should be noted that Biden won the 2020 Election (by most accounts) and that Trump won the 2016 Election. While we do not perform any rigorous statistical tests in this project, our results point to a strong correlation between Twitter sentiment and overall popularity of a candidate.

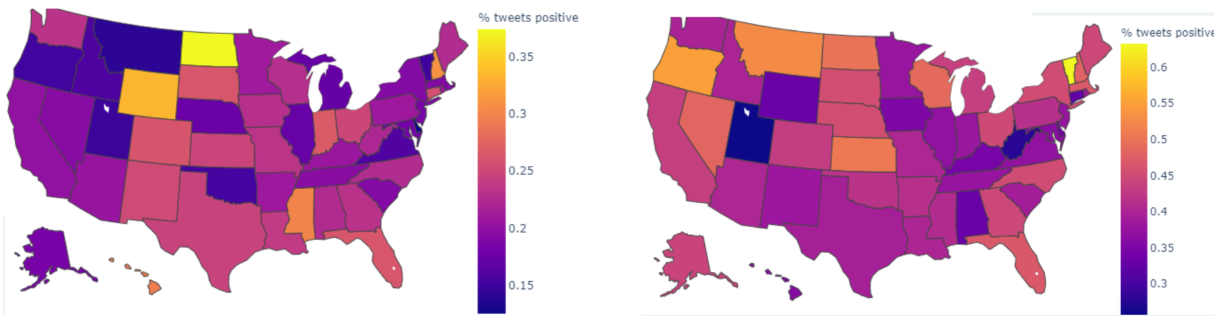
2) Sentiment Toward Election: We were also interested in seeing how Twitter users felt towards the 2020 US Election in general, in addition to the individual party analysis. To do this we generated the same graphs as in the previous section but did not filter for any particular candidate. Thus, we are left with the collective sentiment of all tweets related to the election. Figure 7 shows this sentiment broken down by state. We also performed the same analysis using TextBlob (on 2020 data) which is depicted in Figure 8.

There are some interesting takeaways from these maps. Firstly, sentiment results from TextBlob's NaiveBayesAnalyzer only range from 40% to 55%, which is far less than results from Flair, which range from 25% to 55%. Note how Flair is able to capture the unique negativity exhibited by active Twitter users from Vermont, while TextBlob produced hardly any distinguishable results. The dated Naive Bayes classifier is clearly less able to capture the same range of sentiments expressed on a state-by-state basis as modern context-inclusive NLP model.

Also of note is how the 2020 Election is on average more positive than the 2016 election, with 38.5% of 2020 tweets positive versus 25.8% of 2016 tweets. There are probably countless factors that play into this discrepancy, but one of our interpretations is that 2020 has been a more politically positive year for Democrats than 2016, and that the Twitter-using demographic is generally biased towards the left, resulting in a net positive increase in total sentiment on the platform in 2020 compared with 2016.

D. Candidate Tone Analysis

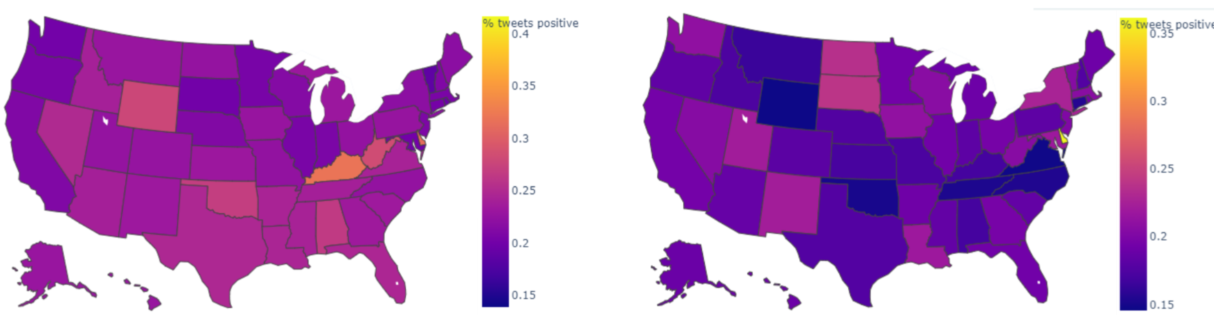
We were also interested in analyzing candidate/party differences in Twitter language usage from an emotional perspective. We use an API for IBM's ToneAnalyzerV3 to break down tweets into their primary emotions, or tone. According to their documentation, IBM uses various features from each sentence such as n-grams, sentiment and more as input into their machine learning classifier. They then trained a support vector



(a) 2020 Twitter sentiment toward Trump

(b) 2020 Twitter sentiment toward Biden

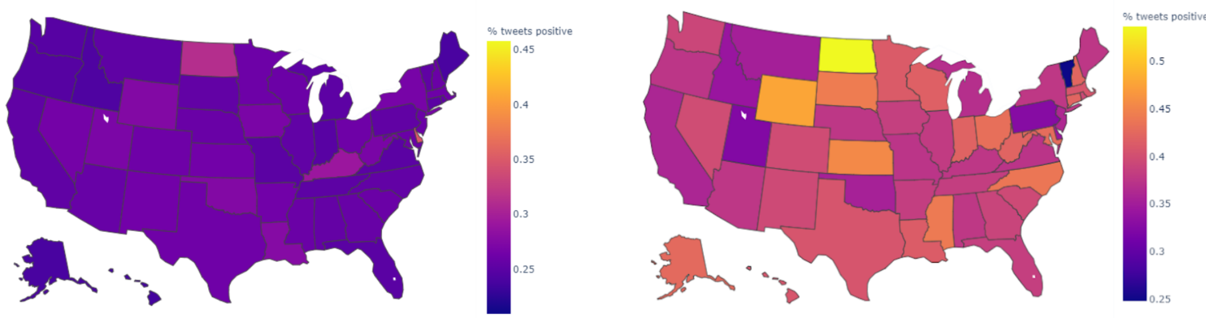
Figure 5: Percentage of positive sentiment Tweets that mentioning Trump or Biden respectively (2020).



(a) 2016 Twitter sentiment toward Trump

(b) 2016 Twitter sentiment toward Hillary

Figure 6: Percentage of positive sentiment Tweets that mentioning Trump or Hillary respectively (2016).



(a) 2016 general election sentiment

(b) 2020 general election sentiment

Figure 7: Percentage of positive sentiment Tweets across US states (both parties included in filter) in 2016 and 2020

machine (SVM) using these features on a large annotated English-language data set.

Using Tone Analyzer we can quantify the emotions used by Twitter users with higher granularity than simple positive or negative sentiment. The IBM models classified based on seven tones: anger, fear, confidence, tentativeness, analytical, sadness, and joy. Due to our limited amount of API credits, we only ran this Tone Analysis on a few key users, namely @realDonaldTrump and @JoeBiden. We also grouped tweets together into batches of 5 in order to conserve our credits. We then ran the tone analysis on the users' timelines in the 6 weeks leading up to the election. For each "document" we send the API, we get a dictionary of tones mapped to the document, along with the weights of the tones. To generate the graphs in Figure 10 and 9 we summed up the tone weights by tone, and divided by the total. This gives us the proportions of tones used by each user.

As

E. Topic Modeling

We used Latent Dirichlet Allocation (LDA) to extract the primary topics from tweets. Our particular implementation of LDA was done via the scikit-learn library in our Python 3 environment. We ran LDA over the tweets from both Trump

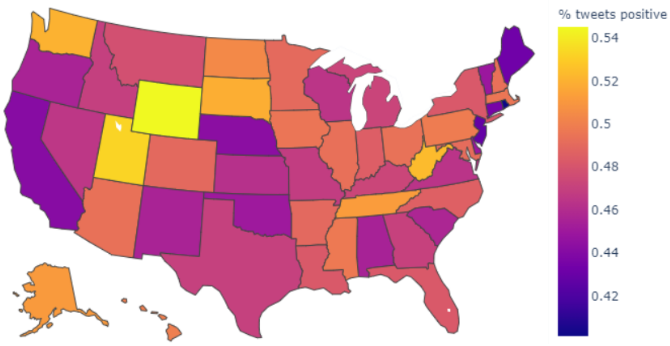


Figure 8: Percentage of positive Tweets across US states in 2020 using Naive Bayes classifier

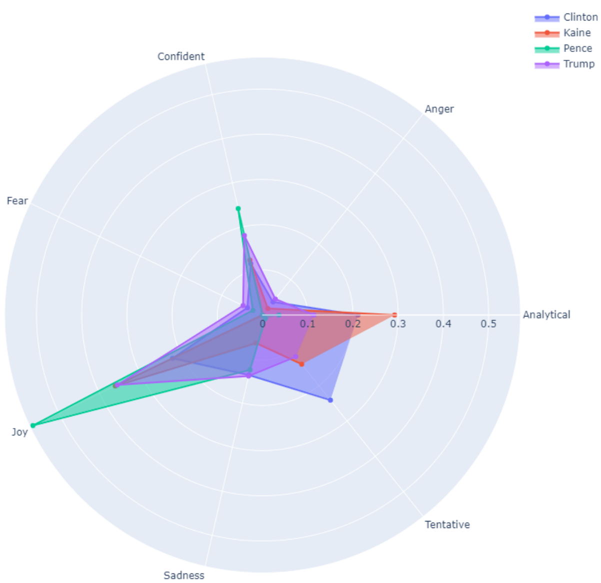


Figure 9: IBM tone analysis for 2016 US Presidential Candidates, click here for an interactive version.

and Biden's Twitter account, as we were interested in which topics each candidate placed emphasis on leading up to the election. If we had more time, we would have expanded our topic analysis to include more politicians, as well as the entire Twitter corpus in election week.

To visualize the results from our topic models, we used word clouds. Word clouds are typically produced from word frequency: the higher the frequency, the larger the word is in the image. We similarly used word clouds to depict the topics extracted with LDA, except the size of a word varies with respect to the LDA weights. We use this visualization technique to analyze differences in word emphasis between topics. Figure 12 and Figure 11 depict the primary topics tweeted by Trump in the weeks leading up to the presidential election in 2016 and 2020 respectively.

Although our topic analysis was limited in scope, it is interesting to note that Hillary was present in relatively more Trump tweets compared with Biden. The main Trump topics

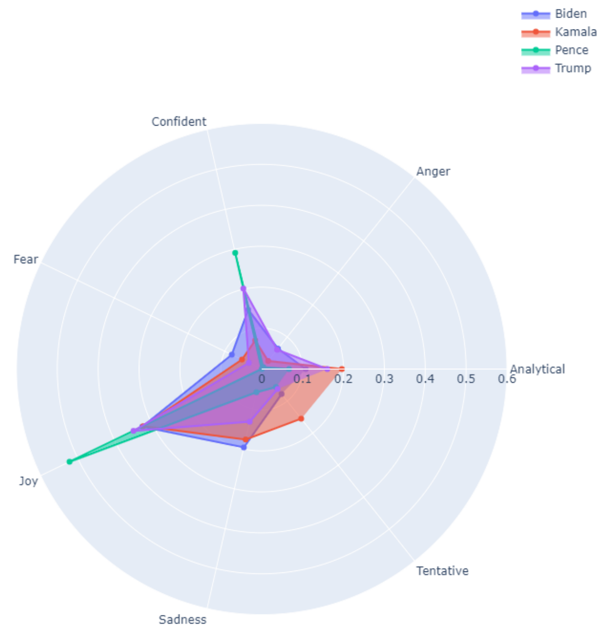
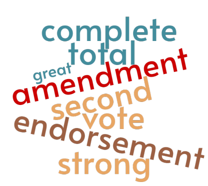
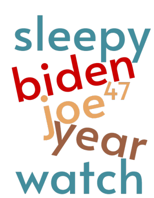


Figure 10: IBM tone analysis for 2020 US Presidential Candidates, click here for an interactive version.



(a) Topic 1

(b) Topic 2



(c) Topic 3

Figure 11: Top three topics extracted from Donald Trump's 2020 Twitter feed in the weeks leading up to the election.

in 2016 also incorporated more vibrant, ad-hominem language compared with 2020, where the main topics were mainly election related.

VI. Conclusion and Future Work

We employed the latest natural language processing tools to analyze Tweets leading up to the 2020 US Election, and contrasted the results with similar analysis of the 2016 election. Although we did not conduct rigorous statistical calculations,

(a) Topic 1

(b) Topic 2

(c) Topic 3

Figure 12: Top three topics extracted from Donald Trump's 2016 Twitter feed in the weeks leading up to the election.

our results nonetheless show key differences between the 2016 and 2020 elections. Results from our sentiment classification and aggregation suggest that overall Twitter sentiment toward the US election is higher in 2020 compared with 2016. We found that, despite the left-shifted demographic of Twitter users, Donald Trump in 2016 enjoyed an overall higher proportion of positive tweets compared with Hillary Clinton, though in 2020 he was less favorable than Biden. Our tone analysis shows that Trump and Pence predominantly use confident and positive language in their Tweets, while Democratic candidates were more well-rounded in their range of tones. Lastly, our topic models show how Trump's Twitter priorities have changed from 2016 to 2020.

There are many ways to expand upon this work in the future. We could incorporate more features into our analysis. How does the sentiment analysis change when weighted by like-count or retweet-count? We would also include more users into our analysis, since the election is more than just the two presidential candidates. Moreover, a more rigorous statistical approach to validate the differences between election years would solidify our findings. A more time intensive direction would be to build a machine learning model to try and predict votes based on historical data, including both federal and state level elections. We believe breakthroughs in natural language processing, such as the contextualized word embeddings we used, will ultimately improve our ability to extract useful information from social media like Twitter.

VII. Breakdown of Work Done

For this project, Collin performed and wrote the sections on the data collection, cleaning, and preprocessing of the data. Collin also performed exploratory data analysis and generated visualizations for the current data by implementing Box Plots, Tree Maps, Density Plots, and Correlograms. Tom implemented and wrote the sections on sentiment analysis, tone analysis, and topic modelling, as well as generating the corresponding visualizations.

References

[MP08] Laura McKenna and Antoinette Pole. "What do bloggers do: an average day on an average political blog". In: *Public Choice* 134.1-2 (2008), pp. 97–108.

[Rao+10] Delip Rao et al. "Classifying Latent User Attributes in Twitter". In: *Proceedings of the 2nd International Workshop on Search and Mining User-Generated Contents*. SMUC '10. Toronto, ON, Canada: Association for Computing Machinery, 2010, pp. 37–44. ISBN: 9781450303866. DOI: 10.1145/1871985.1871993. URL: https://doi.org/10.1145/1871985.1871993.

[Tum+10] Andranik Tumasjan et al. "Predicting elections with twitter: What 140 characters reveal about political sentiment". In: *Fourth international AAAI conference on weblogs and social media*. Citeseer. 2010.

[CM11] Jessica Elan Chung and Eni Mustafaraj. "Can collective sentiment expressed on twitter predict political elections?" In: *AAAI*. Vol. 11. 2011, pp. 1770–1771.

[PP11] M. Pennacchiotti and Ana-Maria Popescu. "A Machine Learning Approach to Twitter User Classification". In: *ICWSM*. 2011.

[ALR12] Faiyaz Al Zamal, Wendy Liu, and Derek Ruths. "Homophily and latent attribute inference: Inferring latent attributes of Twitter users from neighbors." In: *ICWSM* 270.2012 (2012).

[Bor+12] Javier Borondo et al. "Characterizing and modeling an electoral campaign in the context of Twitter: 2011 Spanish Presidential election as a case study". In: *Chaos: an interdisciplinary journal of nonlinear science* 22.2 (2012), p. 023138.

[SCR12] Juan M Soler, Fernando Cuartero, and Manuel Roblizo. "Twitter as a tool for predicting elections results". In: *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE. 2012, pp. 1194–1200.

[Gay13] Daniel Gayo-Avello. "A meta-analysis of state-of-the-art electoral prediction from Twitter data". In: *Social Science Computer Review* 31.6 (2013), pp. 649–679.

[Cal+14] Guido Caldarelli et al. "A multi-level geographical study of Italian political elections from Twitter data". In: *PloS one* 9.5 (2014), e95809.

[Che+14] Justin Cheng et al. "Can cascades be predicted?" In: *Proceedings of the 23rd international conference on World wide web*. 2014, pp. 925–936.

[HCB14] Bo Han, Paul Cook, and Timothy Baldwin. "Text-based twitter user geolocation prediction". In: *Journal of Artificial Intelligence Research* 49 (2014), pp. 451–500.

[Cal+15] N. A. Calderon et al. "Mixed-initiative social media analytics at the World Bank: Observations of citizen sentiment in Twitter data to explore "trust" of political actors and state institutions and its relationship to social protest". In: *2015 IEEE International Conference on Big Data (Big Data)*. 2015, pp. 1678–1687.

[Eic+15] Johannes C Eichstaedt et al. "Psychological language on Twitter predicts county-level heart disease mortality". In: *Psychological science* 26.2 (2015), pp. 159–169.

[LWK16] Justin Littman, Laura Wrubel, and Daniel Kerchner. *2016 United States Presidential Election Tweet Ids*. 2016.

[Pre+17] Daniel Preoţiuc-Pietro et al. "Beyond binary labels: political ideology prediction of twitter users". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2017, pp. 729–740.

[Yaq+17] Ussama Yaqub et al. "Analysis of political discourse on twitter in the context of the 2016 US presidential elections". In: *Government Information Quarterly* 34.4 (2017), pp. 613–626.

[ABV18] Alan Akbik, Duncan Blythe, and Roland Vollgraf. "Contextual String Embeddings for Sequence Labeling". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1638–1649. URL: https://www.aclweb.org/anthology/C18-1139.

[Dev+18] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[YMO18] Xiao Yang, Craig Macdonald, and Iadh Ounis. "Using word embeddings in twitter election classification". In: *Information Retrieval Journal* 21.2-3 (2018), pp. 183–207.

[VP20] Georg Velev and Iliyana Pekova. "Analysis of social media behavior of the 2020 presidential election candidates". In: *Machine Learning* 3 (2020), p. 1.