Exploring Polling Data and Twitter Feeds with TFIDF Analysis

Introduction

The 2020 presidential election was a topic of interest for most of this year. Exploring political polling data as well as Twitter data seemed like an interesting and appropriate approach as Twitter has been a major platform for politicians to communicate to the public. Extracting tweets from twitter and polling data from FiveThirtyEight allowed us to simultaneously look at trends in polling approval and what candidates tweeted as polling data changed. Through analysis using Python, we can infer potential relationships between approval ratings and important topics discussed by the candidates on Twitter. As a group we used a text analysis method called TF-IDF to estimate word importance from tweets. Using this text analysis method along with visualization packages in Python, our group extracted trends from Twitter and polling data and combined them into visualizations that provide insights to the 2020 presidential election.

Data

The polling data came from FiveThirtyEight's collection of political polling data[1], which we chose for it's availability, reliability, and transparency as a data source. The specific dataset we used, president_polls.csv, is a collection of general election polls organized by date, pollster, candidate, and vote percentage (more details about the dataset provided in the appendix). This dataset covers 777 separate polls between 2018-11-27 and 2020-11-01, including primary and independent candidates, across 35 states and three electoral districts (Maine CD-1 and CD-2, Nebraska CD-2). FiveThirtyEight also assigned a reliability grade (A+ to F) to each of the 36 pollsters included in the dataset.

To prepare this data, we first included only polls taken after the Democratic primaries were effectively over (we defined this as the moment polls stopped including Bernie Sanders as a potential candidate). We chose to only include polls with a B or higher reliability rating from FiveThirtyEight to avoid overly biased polls. We then constructed a runoff pivot table which averaged the candidates score across all polls taken by day, weighting it by the sample size of the respective polls. This was done to ensure that larger polls had more bearing on the overall

[1] https://data.fivethirtyeight.com/

candidate rating for a given day. We also constructed a weekly rolling average to get a better understanding of the overall movement of the data.

To access Twitter data for both presidential candidates, we initially looked into Twitter's official API, however the limitations placed on downloads would not give us adequate data for our project. Instead we used an open source web scraper called Twint (2020). The package can be accessed on github and can query tweets by user. In total we scraped approximately 54,000 tweets from Donald Trump's Twitter account and approximately 6,200 tweets from Joe Biden's Twitter account. We isolated tweets from April of 2020 through November 1, 2020 because this was around the time that official candidates were chosen for the general election. Our final analyses included only this time frame for both candidates.

To make tweets easier to process and more interpretable, we removed the following text: URLs, numeric characters, and punctuation. We also removed less informative words like English stop words and any word with 2 or fewer characters.

Text Analysis: TFIDF

The key challenge in trying to extract information from text is converting natural language into a useful numerical quantity that reflects its meaning. In our case, we are interested in looking at short documents (tweets) and coming up with some metric that can recognize information rich terms and describe language trends of a specific document within the context of the document space. Such a metric would allow us to explore and compare tweeting trends in a quantitative and rigorous way. Strategies that have been designed precisely for these types of problems can range from simple counting algorithms such as Bag-of-Words to more complicated processes such as word-to-vec modeling [bag of words, word to vec].

Given the scope and purpose of this project, we propose using a TF-IDF (Term Frequency, Inverse Document Frequency) method to analyze tweet documents. TF-IDF is a relatively heuristic extension of the Bag-of-Words method that represents documents as a vector indexed by each unique term appearing across all documents (the set of all unique terms is called the corpus) [TF-IDF]. The strategy is designed to first assign a numerical score to each unique term in a document corresponding to the frequency of a term within that document (called the Term Frequency) and then assign each unique term from the corpus with a score that is inversely proportional to its frequency across all documents (called Inverse Document Frequency, see appendix). Thus the final TF-IDF score vector can be determined by taking a sum of the term frequency scores weighted by their corresponding inverse document frequency scores. Consequently, terms that are common across all documents are given little weight and are considered less meaningful for representing a particular document, while rare words are considered far more important in representing the document. In the context of our project, the TF-IDF score for each term in a tweet offers a quantitative ranking of how informative that word is.

Using fundamental Python tools, we were able to build a TF-IDF scoring algorithm from scratch and apply it to our tweet data. To calculate the term frequency we start by tokenizing the document into individual terms and using a dictionary-like object called a “counter” to associate each unique term within a document with its frequency [counter objects]. Next, we calculate the Inverse Term Frequency of each term (Appendix Equation 1) by taking the inverse log of the amount of documents that contain that term. Again we use a counter object to associate each term in the corpus with its inverse document frequency score. Unlike the traditional Python dictionary object, counter objects allow for vector operations with respect to keys, making it a preferable choice for representing term frequency arrays. With both term frequencies and inverse document frequencies “vectorized” in counter objects, vector multiplication gives the TF-IDF scores for each tweet.

Results

The primary results of this project are the TF-IDF scores and the trend of these scores overlaid on polling data. These figures were created using matplotlib’s pyplot. Figure 11 shows what the TF-IDF scores look like on a certain day. In this case, Biden’s top three words on August 11th, 2020 were “win,” “go,” and “kamalaharris,” his running mate. These scores give insight into what was important on that day.

TF-IDF can also be used to look at the tweeting trends of a month. Figures 5 - 10 show the TF-IDF scores for each month’s tweets from August through October for both Donald Trump and Joe Biden. These figures give insight into what each candidate was tweeting about most often in a particular month. For example, in August Joe Biden’s top word by TF-IDF score was “demconvention,” signifying that during the month of August Biden’s most informative tweets had concerned the upcoming democratic convention.

Combining the TF-IDF calculations with the polling data gives the results shown in Figure 1 and Figure 2. Figure 1 shows the raw polling data percentages overlaid with top TF-IDF scoring words for days where there was a sudden change. In this case, a sudden change was defined as a change in 5 percentage points or more in a single day. Some of these words can give insight into the reason why the changes happened, such as Biden's large spike in August corresponding to "demconvention." In September, there was a drop in polling for Trump corresponding to "harry," potentially referring to when Prince Harry encouraged Americans to vote for Joe Biden. In October and November, there are spikes for Trump for "andrewmccarthy" and "voteearlyday" respectively. These correspond to events where Andrew McCarthy defended Trump about impeachment and vote early day, which gave hope for Trump's election chances. Overall, this figure gives insight into the reasons why certain days may have large changes in polling, but not any insight towards the overall trend.

Figure 2 shows the overall trend of polling much more clearly, because it consists of the rolling average of polling data on a weekly basis. This polling data is overlaid with top TF-IDF scoring words for certain days randomly dispersed across the chart. This chart shows the ups and downs of polling as well as what was being tweeted at that time by the candidates. For example, in mid-October there is an upward spike for Joe Biden in approval percentage, coinciding with "kamala harris" as having the top TF-IDF score.

Conclusion

During an election season, Twitter provides both a space to campaign in and a barometer of the current political mood and focus. In light of this, our team thought the twitter accounts of the two 2020 presidential candidates were ample test sets for TF-IDF analysis. With a baseline TF-IDF algorithm, we were able to extract some basic information about tweeting trends and explore the relationship between presidential tweets and the more conventional political metric, polling. It is important to stress that the purpose of this project was not to infer causal relationships between tweeting, polling, and political events, but to build from scratch the infrastructure necessary to begin these questions. Looking forward we note several improvements could be made if the goal was purely research focused, such as using a more elaborate text analysis algorithm or processing text in more sophisticated ways. The project, however, confidently demonstrates the usefulness of even a basic TF-IDF on what can be a particularly thorny topic to parse: political discourse.

Works Cited

Our Data. (2018, February 9). FiveThirtyEight. https://data.fivethirtyeight.com/

T. (2020). *twintproject/twint*. GitHub. https://github.com/twintproject/twint

Appendix

$$IDF(term) = log\left(\frac{\text{Total Tweet Count}}{\text{Number of Tweets Containing "term"}}\right)$$

Equation 1: IDF scores

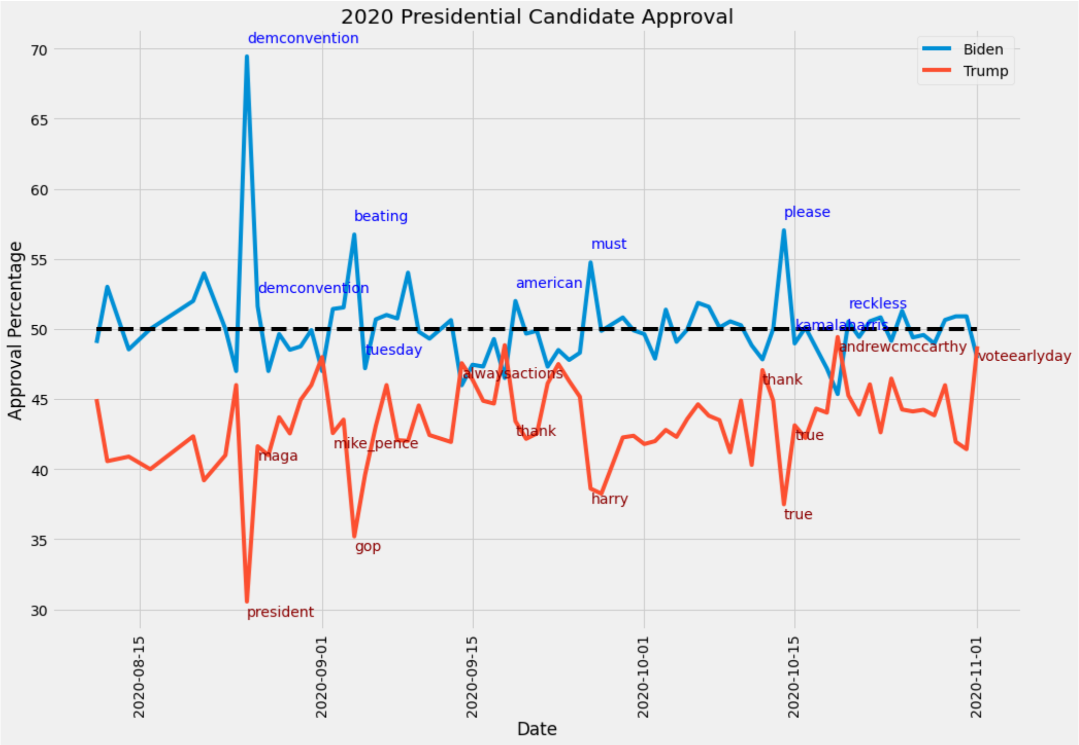

Figure 1: Daily Polling Data Overlaid with Top TFIDF Scoring Words

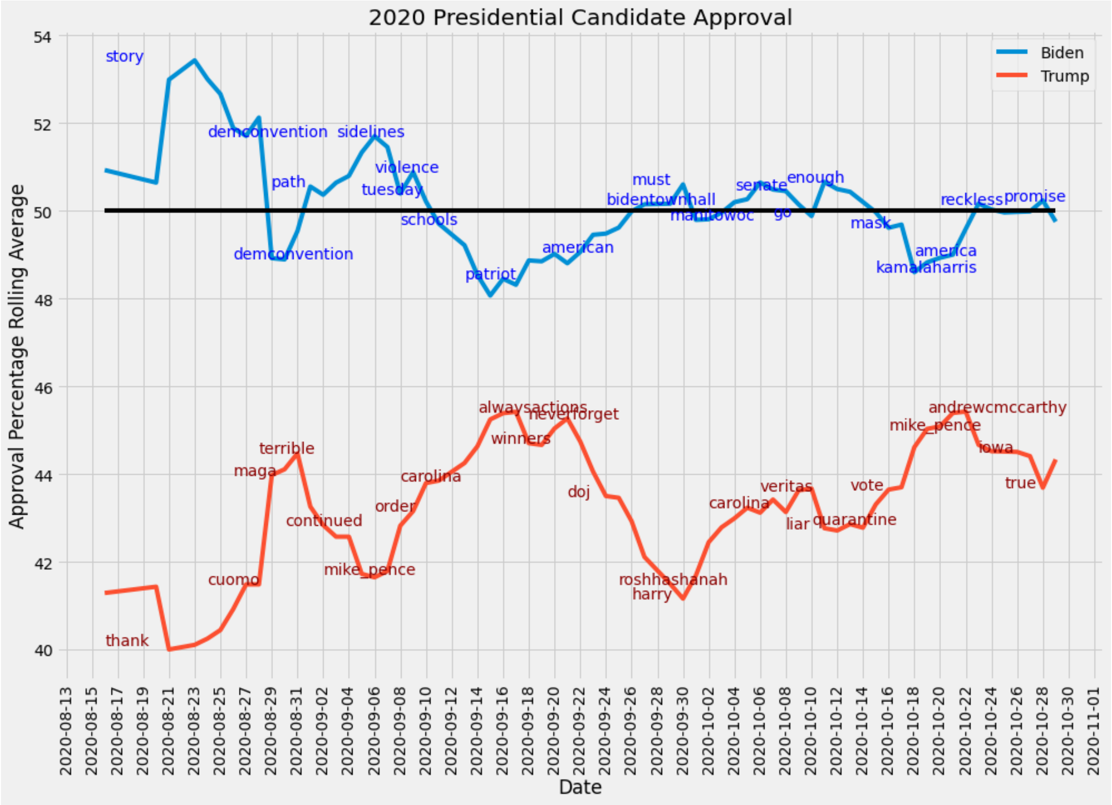

Figure 2: Rolling Average Polling Data Overlaid with Top TFIDF Scoring Words

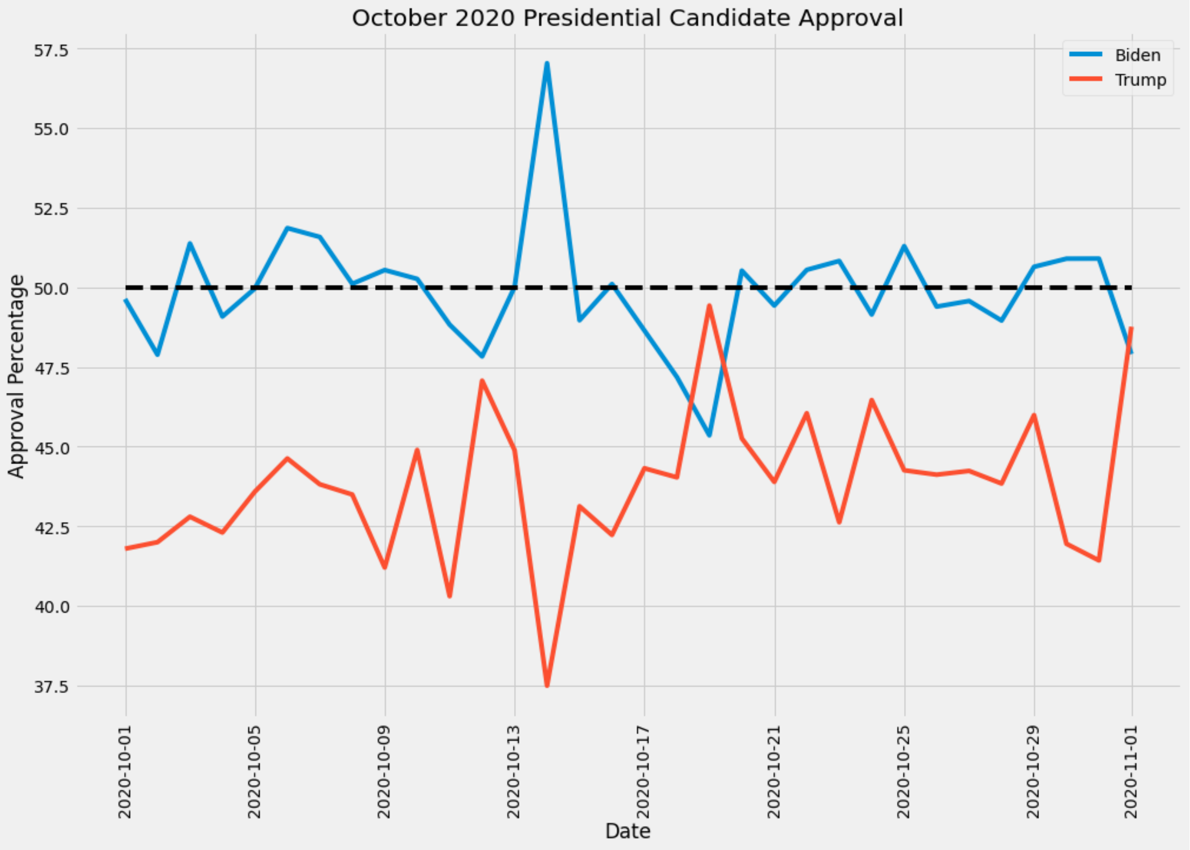

Figure 3: October 2020 Presidential Candidate Approval

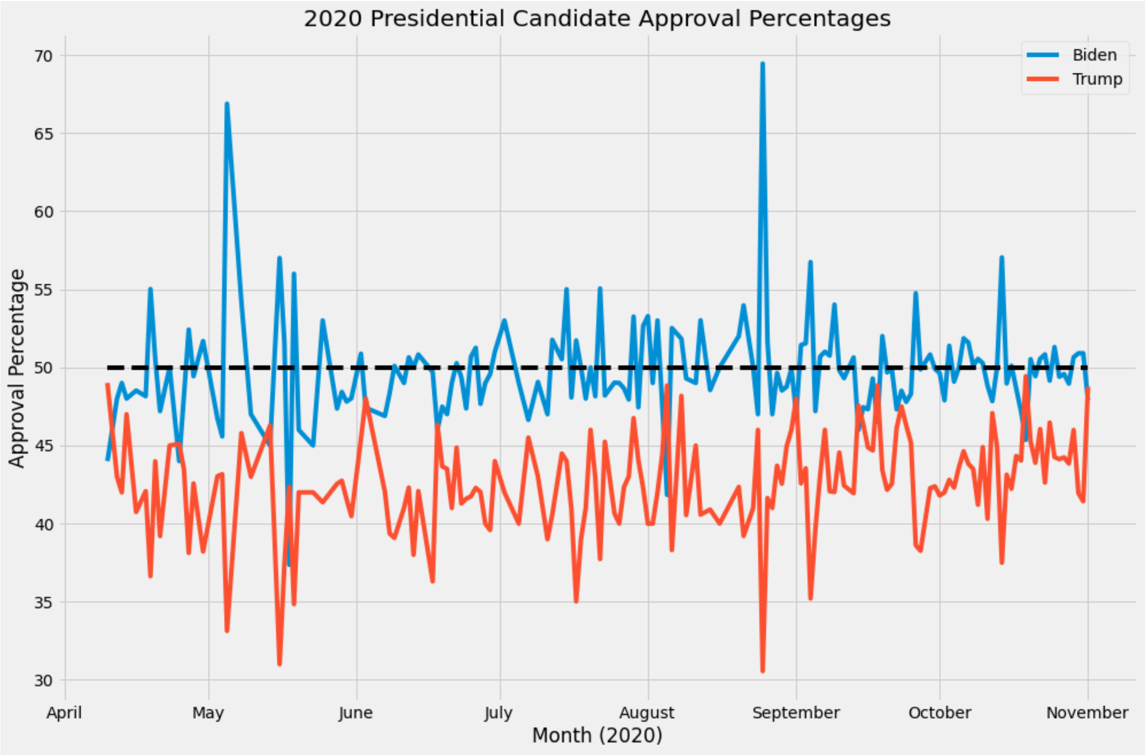

Figure 4: Overall 2020 Presidential Candidate Approval

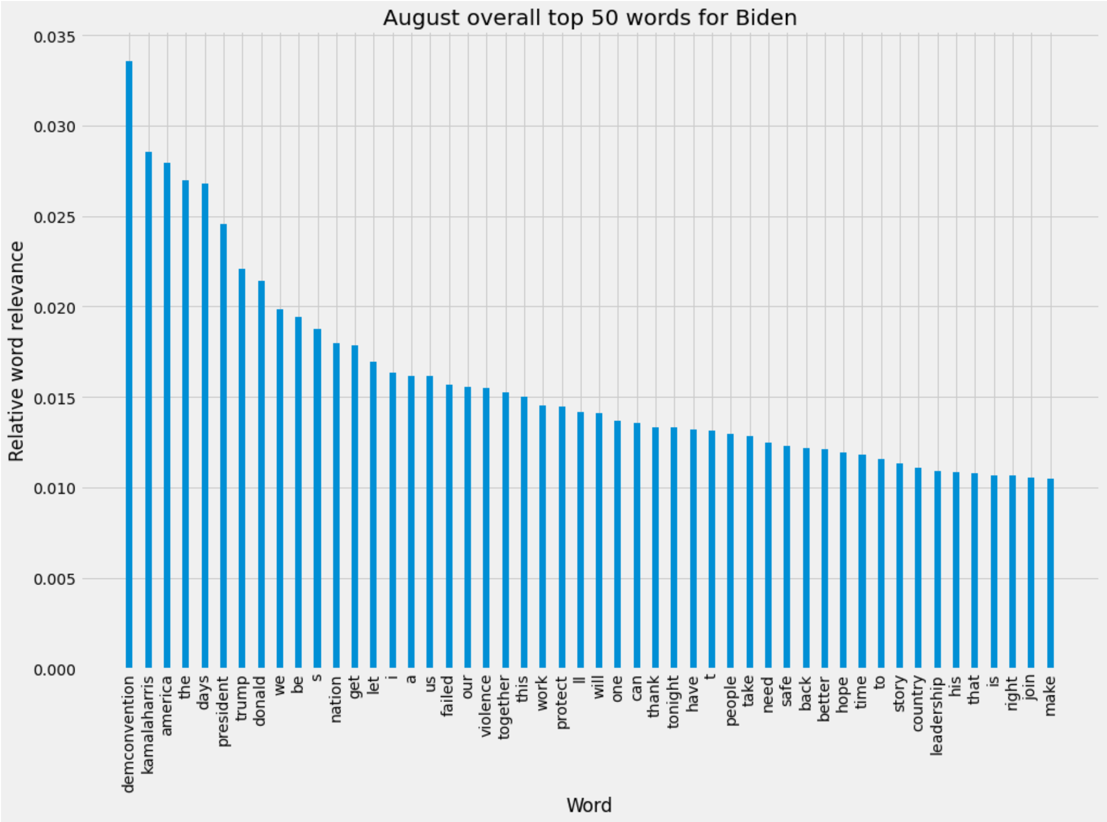

Figure 5: August Top 50 TFIDF Scoring Words for Biden's Tweets

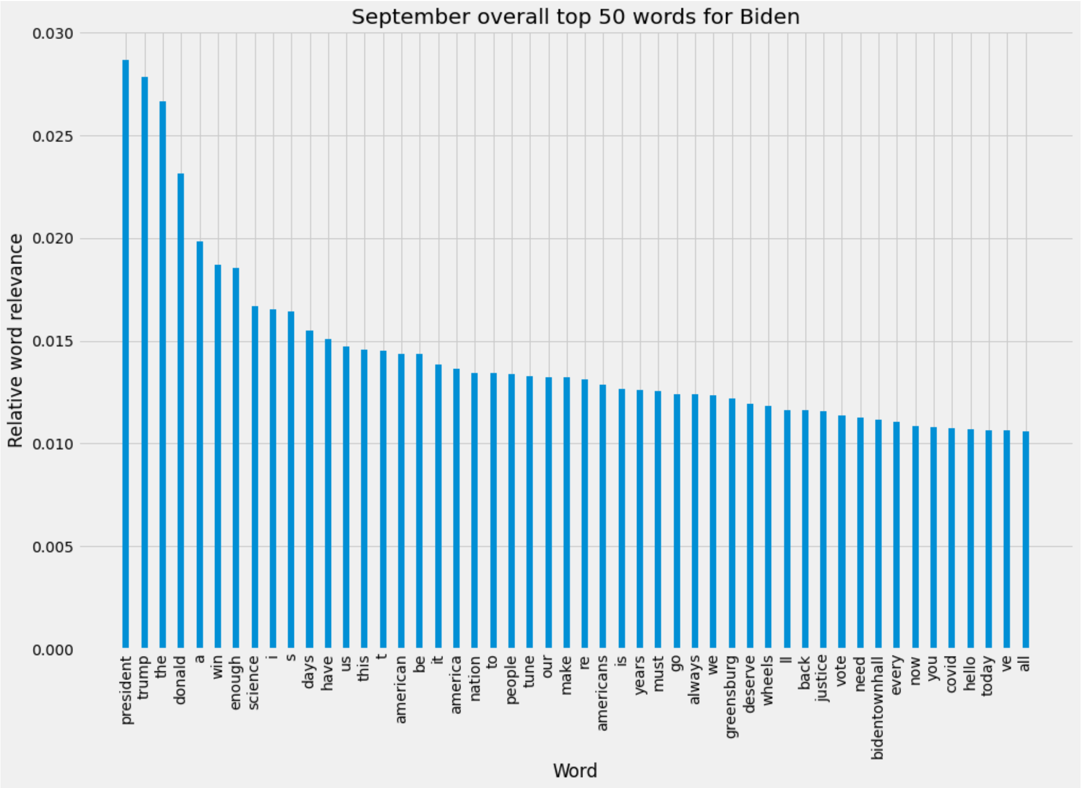

Figure 6: September Top 50 TFIDF Scoring Words for Biden's Tweets

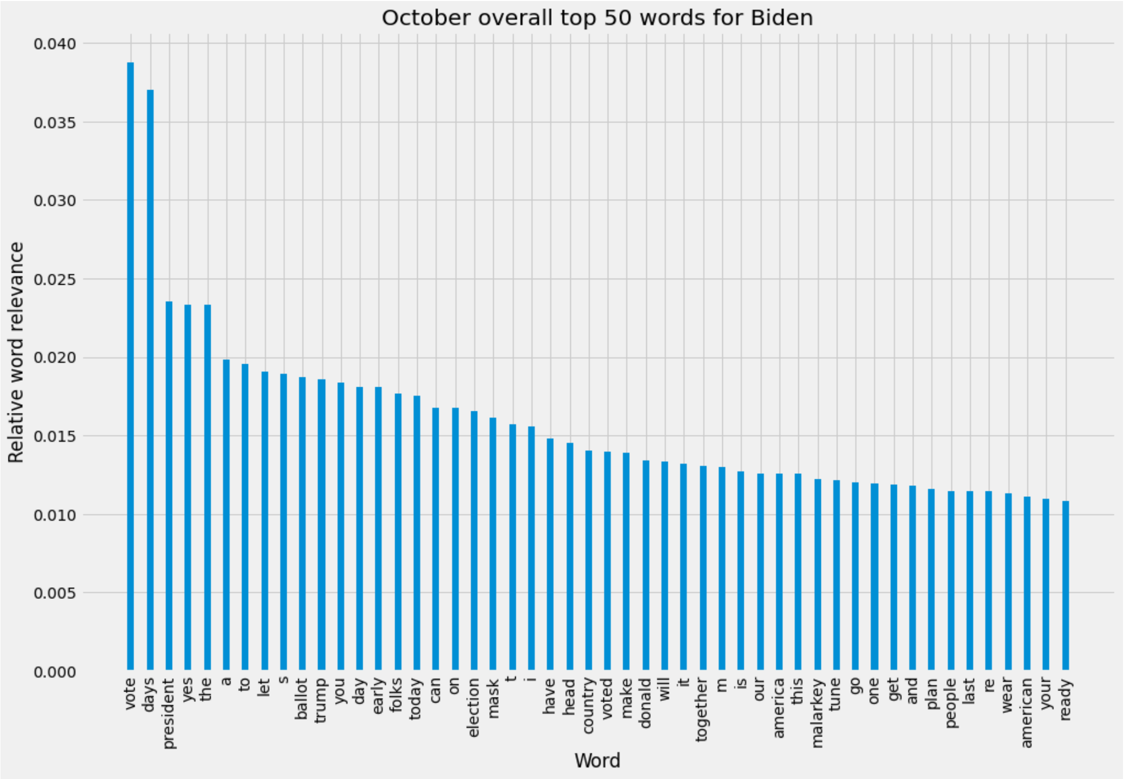

Figure 7: October Top 50 TFIDF Scoring Words for Biden's Tweets

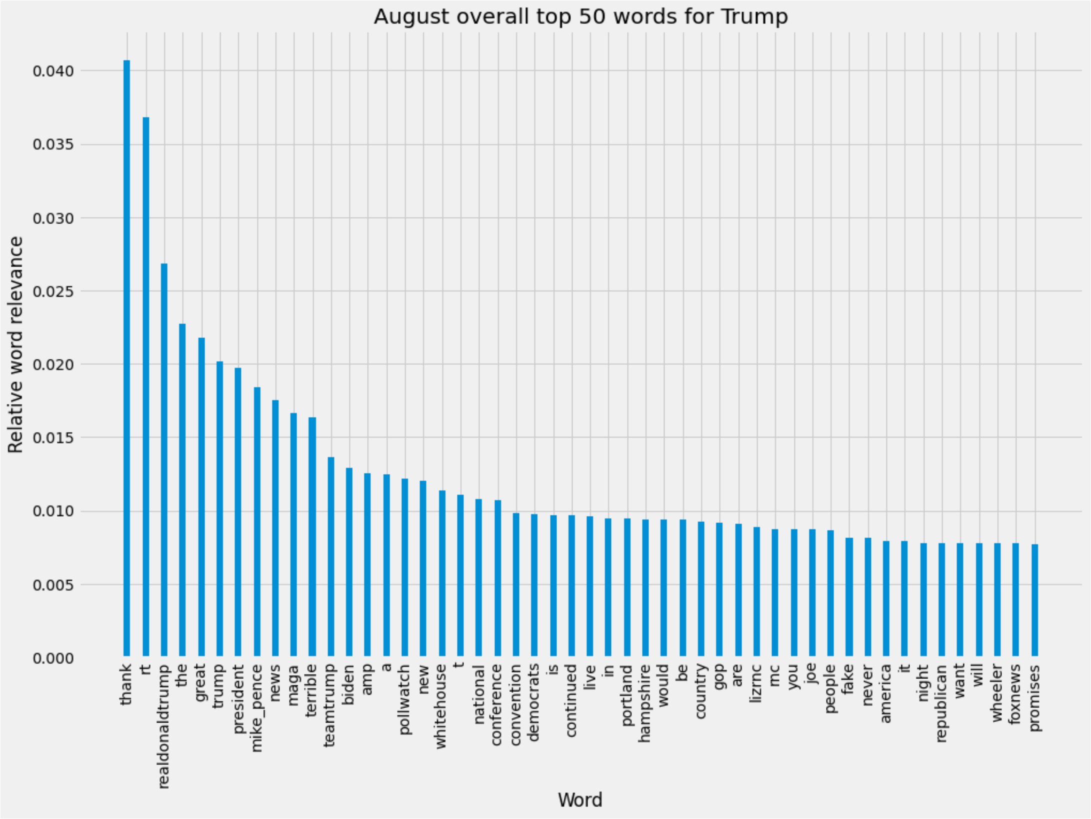

Figure 8: August Top 50 TFIDF Scoring Words for Trump's Tweets

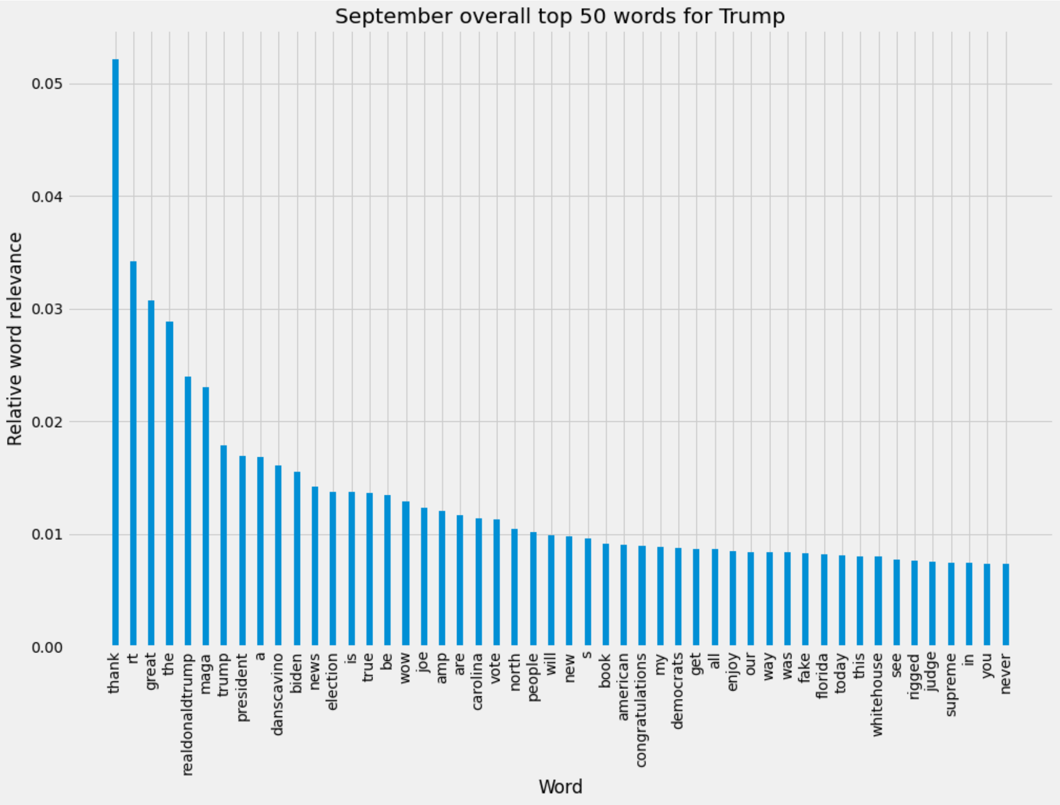

Figure 9: September Top 50 TFIDF Scoring Words for Trump's Tweets

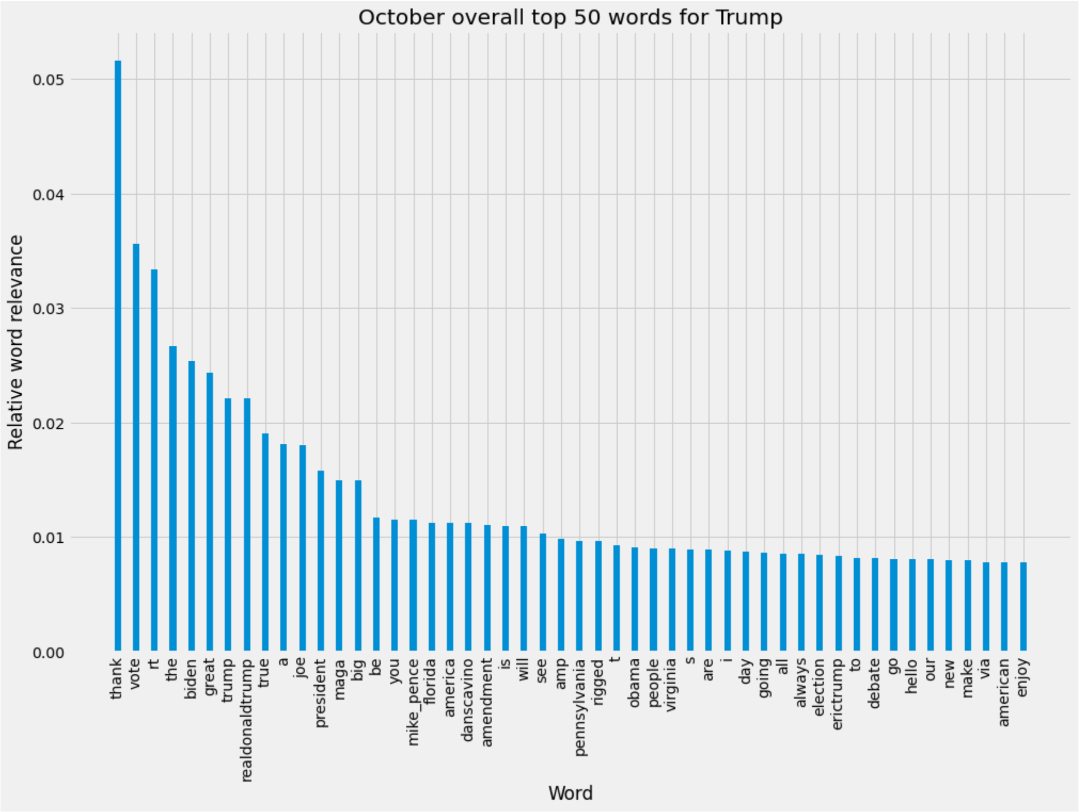

Figure 10: October Top 50 TFIDF Scoring Words for Trump's Tweets

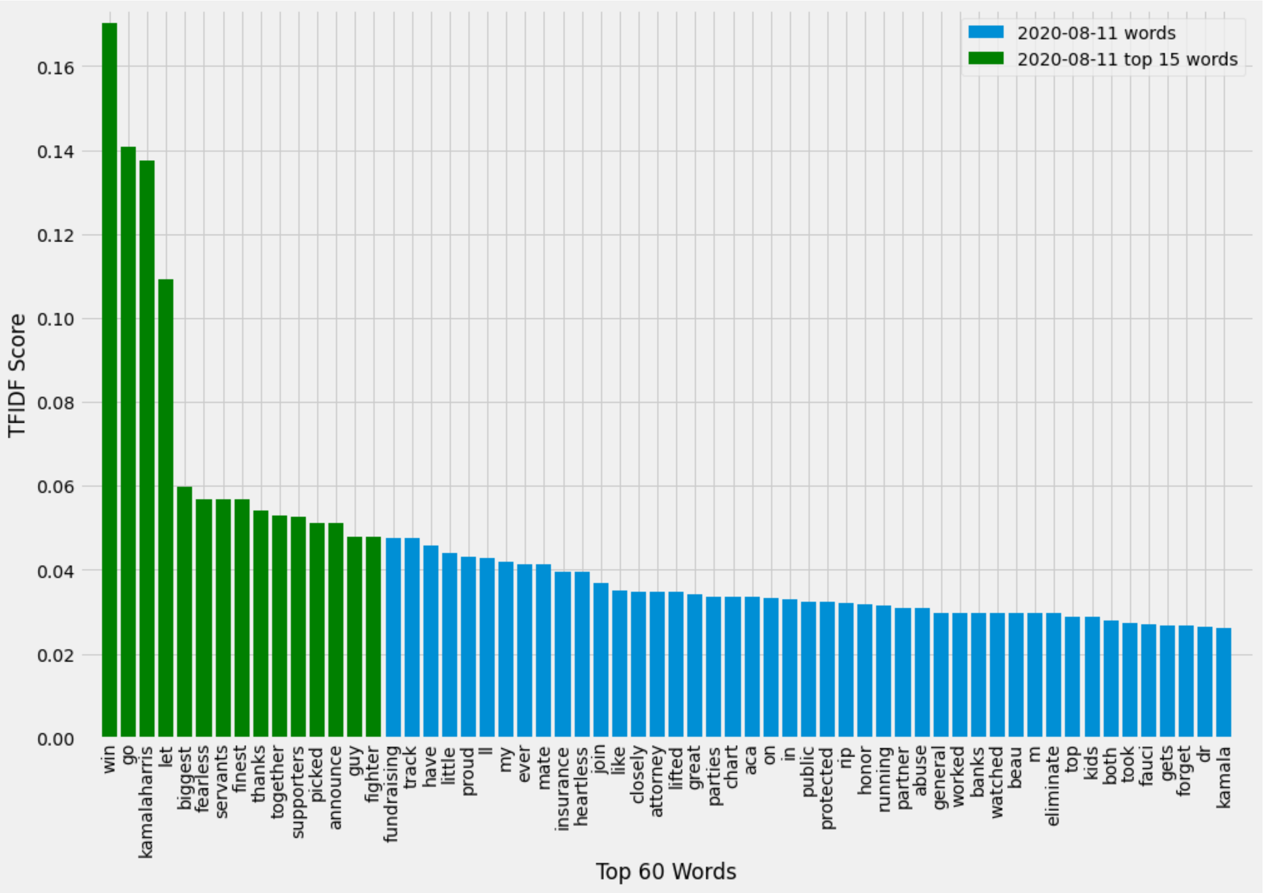

Figure 11: Top 60 TFIDF Scoring Tweet Words for Joe Biden on August 11th, 2020