Twitter Sentiment of Donald Trump and Barack Obama

Maheen Asghar
School of Information
University of Michigan
masghar@umich.edu

Dorsa Massihpour
Data Science
University of Michigan
dmassihp@umich.edu

December 13, 2019

Abstract

In the past decade, there has been much interest in the area of sentiment analysis, especially with regards to tweets. In our research, we focus on analyzing and predicting the sentiment of the past two presidents, Donald Trump and Barack Obama. Using three types of different classifier, we predict whether a tweet is positive, neutral, or negative, and evaluate our models using the F1 score and accuracy measures. The dataset used for training our models was taken from data.world and included tweets spanning from 2009 to 2016, when Trump was elected president. We use these predicted tweets to conduct deseasonalized time series analysis on the frequency of tweets for each president to explore specific time periods and the majority tweet sentiment for each president, as well their overall sentiment. We find that the Logistic Regression was the best-performing in both evaluation measures, for both presidents. Also, Trump's tweets tend to be mostly positive before 2015, after which they become mainly neutral, while Obama's tweets maintain their predominantly neutral sentiment throughout this time period.

1 Introduction

Sentiment analysis, also known as "opinion mining" is the process of identifying and categorizing pieces of text according to its overall opinion. It comprises one branch of the field of natural language processing, which is focused on helping computers understand and manipulate human language. Subjectivity/objectivity identification and feature/aspect based sentiment analysis are the two most common types of sentiment analysis. The former aims to classify a group of words, such as a sentence, as being either objective or subjective, and the latter groups components of an entire unit of text into different sentiment groups.

Over time, Twitter has grown into a platform where users can post messages about their opinion on a variety of topics, discuss current issues, complain about products or people. It's especially been utilized by the current and previous presidents: Donald Trump and Barack Obama. From Trump's official declaration of candidacy in 2015 through the first two years of his presidency, he tweeted over 17,000 times, while Obama tweeted less than that during the entirety of his time in office. The content of Trump's tweets have been widely debated, however the analysis on the style of Trump's tweets has been limited to misspellings and grammar mistakes. With the rise of use of social media by the current president, it begs the question–what is the sentiment of our current president? Does it differ from the previous president? In this paper, we perform sentiment analysis on their tweets to determine what each president's style is like.

There are various types of sentiment analysis: document-level, sentence-level, and aspect-level. In document-level sentiment analysis, each document focuses on a single-entity and comes from a single opinion holder and from here, the opinion can be classified as positive, negative, or neutral. Alternatively, in the sentence-level approach, each sentence is assigned their own sentiment. Our research takes a similar approach, but for tweets, which are limited to 280 characters each. Furthermore, we adopt a lexicon-based technique to determine the sentiment of the tweet, looking at the semantic orientation of words within a tweet. While we assign initial sentiment using the TextBlob package, we build a series of models that use the order, frequency, and vocabulary of words to predict a tweet's overall sentiment. As our goal was to not only gauge the overall tweet sentiment of the current and previous president, but also to pinpoint specific times in which the frequency of their tweets for a specific sentiment spiked, we also analyzed the two presidents' time series charts. We found that the Logistic Regression classifier performed the best, having the highest accuracy and F1 scores, and that the two presidents differed in both their overall tweet sentiments and consistency of their tweet sentiments.

2 Related Works

This type of analysis originated in the 20th century, but 99% of research papers in this field were written after 2004. Although there are many published papers on the twitter sentiment analysis of presidents, there are very few studies comparing Trump and Obama specifically, and simultaneously analyzing the frequency of their positive, negative, and neutral tweets over time.

Čišija et all [4] investigated the sentiment of Twitter users during Donald Trump's presidency from 2012 to 2018 using the AYLIEN Text Analysis API. They utilized a similar form of preprocessing, involving tokenization, stopping, stemming, text normalization. After the sentiment analysis, they generated two types of results: sentiment overview and individual clusters word overview. Overall sentiment overview showed that most tweets expressed a neutral opinion, followed by a negative opinion. The word overview showed more negative words, meaning that 'Trump' was more likely to be associated with a negative connotation.

Yaqub et al [9] conducted an analysis of political discourse on twitter during the Presidential Elections in 2016. They investigated the sentiment of tweets of two then presidential candidates: Hilary Clinton and Donald Trump, along with 3 million tweets from Twitter users. After their preprocessing steps, they used SentiStrength to determine the sentiment of tweets. They discovered that sentiment and topics expressed by Twitter users is a good proxy of public opinions. They also found that Donald Trump expressed more positive sentiment in his tweets than Hilary Clinton did and generally had better sentiment in mentions by other users.

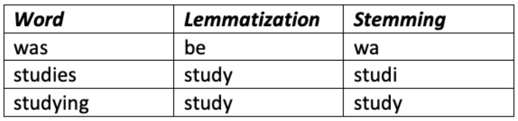
3 Methods

3.1 Data

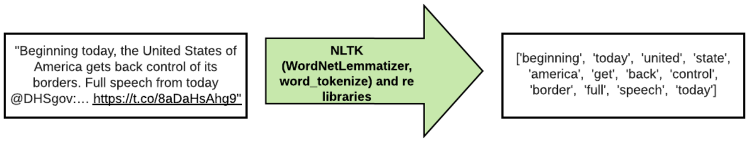
The data we used to evaluate our models came from data.world. The datasets for each of the president spanned a different number of years, but for uniformity we chose to take the subset of tweets from 2009-2016 for both presidents. After extracting the data from these years, Trump had 30,385 tweets and Obama had 27,667 tweets. Both datasets contained the tweet, along with the timestamp, but Trump's dataset also included the device it was tweeted on while Obama's dataset included the url of the tweet and various tweet interactions, such as retweets and reply to statuses.

3.2 Preprocessing and Evaluation Methods

In order to prepare our date for training on our classifiers, we put them through a series of preprocessing steps. After removing all rows with missing timestamps, we combined the files containing tweets from different years for each president, so that we had one dataframe for each president. We extracted the tweet column from each of the dataframes, converting them into two separate lists and processed them. Using Python's nltk library, we tokenized the tweets into their component words and removed emoticons, symbols, pictographs, map symbols, flags, and stopwords. For this part, we followed a pre-processing article published on the website Towards Data Science. Next, using Python's re library, we removed punctuation and numbers, and converted all tweets into lowercase to make parsing easier. Finally, we performed lemmatization on the two lists of tweets using WordNetLemmatizer in Python, which reduced each word into its common base word, known as the lemma. Unlike stemming, lemmatization follows a procedure which produces an actual language word. We chose lemmatization over stemming for readability of the processed tweets. This distinction is highlighted in the table below, using the words “was”, “studies”, and “studying”. As another example, through lemmatization, the words “learns”, “learned”, and “learning” would all be converted to their base word, “learn”. An example of the raw tweet, and the processed version is shown below.



Word	***Lemmatization***	***Stemming***
was	be	wa
studies	study	studi
studying	study	study



"Beginning today, the United States of America gets back control of its borders. Full speech from today @DHSgov:... https://t.co/8aDaHsAhg9"

NLTK (WordNetLemmatizer, word_tokenize) and re libraries

['beginning', 'today', 'united', 'state', 'america', 'get', 'back', 'control', 'border', 'full', 'speech', 'today']

3.3 Sentiment Analysis

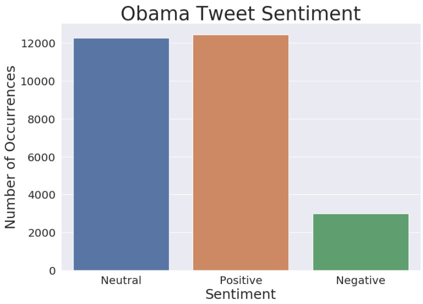
To determine the sentiment of the tweets after preprocessing, we utilized Python’s Textblob package, which returned polarity and subjectivity of the overall tweet. For example:

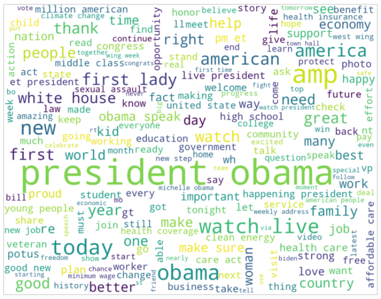
```
TextBlob("Just tried watching Saturday Night Live - unwatchable! Totally biased,
    not funny and the Baldwin impersonation just can’t get any worse.
    Sad").sentiment
Sentiment(polarity=-0.17090909090909093, subjectivity=0.77)
```

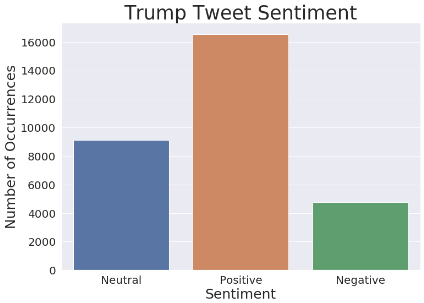
This tells us that Trump’s tweet has a polarity of -0.17, indicating that it slightly negative, and a subjectivity of 0.77, meaning that it is fairly subjective.

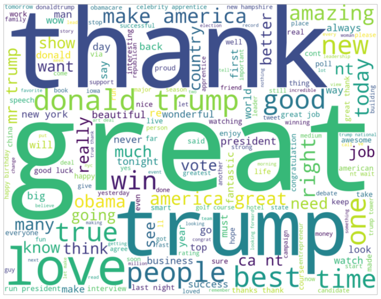
```
TextBlob("bad saturday night live worst nbc not funny cast terrible always complete
    hit job really bad television").sentiment
Sentiment(polarity=-0.4698051948051947, subjectivity=0.7476190476190476)
```

Taking a look at the results again after it has gone through the preprocessing steps, we see the polarity is lower, meaning it is definitely a negative tweet. For our models, we used polarity to label our dataset of tweets.









When looking at Trump versus Obama's tweets, we found that Obama had a more even distribution of positive and neutral sentiment of tweets, while having very little negative sentiment tweets. Trump had a disproportionately higher number of positive sentiment tweets over the other categories.

3.4 Machine Learning Models

The four machine learning models we used were from the scikit-learn's library and included naïve Bayes, logistic regression, random forest, and support vector machine.To actually use our tweets in the classifier, we need to convert the words into numbers using the CountVectorizer library, which takes the processed tweets and counts the frequency of the words per tweet to create a bag of words. However, to account for words that occur frequently, but don't contain useful information, we need to lower their importance using the TF-IDF Vectorizer. TF-IDF (term frequency-inverse document frequency) is a method that increases the weight of a word proportionally to the number of times a word appears in a tweet, but is offset by the number of tweets that contain that word. The vectorizer, transformer, and classifier are then put into a pipeline that behaves as a compound classifier, where we can then do parameter tuning with gridsearch cross fold validation.

```
parameters_multinomial_nb = {
  'bow__ngram_range': [(1, 1), (1, 2)],
  'tfidf__use_idf': (True, False),
  'clf__alpha': (1e-2, 1e-3),
}

parameters_random_foest = {
  'bow__ngram_range': [(1, 1), (1, 2)],
  'tfidf__use_idf': (True, False),
  'clf__n_estimators': (10, 50),
}
```

To evaluate our models and choose the best-performing one, we used two measures, the F1 score and the common measure of accuracy. While accuracy measures the number of data points that are correctly classified, the F1 score is a weighted average that takes into account both the recall and precision. As is standard procedure, we chose to use the F1 score as well since a model can have a high accuracy if it always predicts one outcome, but also have zero predictive power. However, precision and recall measure the fraction of data points that were truly correct out of all data points that the classifier labeled as positive, and, out of all the positive data points, how many the classifier predict correctly, respectively. Thus, by taking into account the number of false positives and false negatives, the F1 score provides a more meaningful evaluation metric. The higher the F1 score, the better the predictive ability of the classifier.

$$Accuracy = \frac{(Tp + Tn)}{(Tp + Tn + Fp + Fn)}$$

Where Tp, Tn, Fp, and Fn are the number of true positives, true negatives, false positives, and false negatives, respectively, based on the model's predictions.

$$F1 = 2 * \frac{(precision * recall)}{(precision + recall)}$$

Where recall is the number of true positives divided by the sum of true positives and false negatives, and precision is the number of true positives divided by the sum of true positives and false positives.

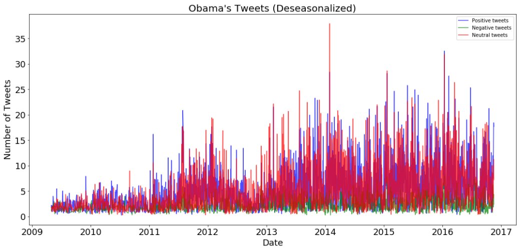
3.5 Time Series

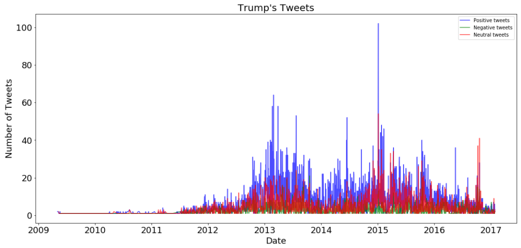
For our time series analysis, we used the sentiment column generated from the Textblob package, along with the timestamp column of our dataframe. We first created buckets representing each unique data by extract the data as a string from the timestamp, and counting the number of tweets for each date for each of the three sentiments. We now had a dataframe with 3 columns representing the sentiment, and whose rows consisted of dates. With the StatsModels package in Python, we used a centered moving averages method to decompose each group of positive/negative and neutral tweets into three components: the trend or long-term movements, the seasonality or repeating short-term patterns, and random variation or noise. (Note that these are the specific components that are returned by the python function that we used. (seasonal_decompose). We used a frequency of 365, as we had tweets spanning almost every day of each year (though some days had multiple tweets), and a multiplicative model as opposed to an additive model for the following reason: the additive model assumes that the magnitude of seasonality does not change with time, wheras in multiplicative models, the magnitude of the seasonal pattern increases or decreases. When plotting the raw time series data for each president, we were able to see that the latter was the case, as seen in the non-deseasonalized time series plots below. We extracted the seasonal component and divided each entry of our created dataframe by these seasonal components. Our reason for deseasonalizing the data was so that we could more clearly view long term trends within the period of 2009 to 2016. Since we are considering a time period in which Obama was president, and Trump was a presidential candidate, we wanted to ignore any cyclical patterns and focus on more meaningful changes unique to these years, such as whether a spike or sudden decrease in negative tweets was due to a specific political event.

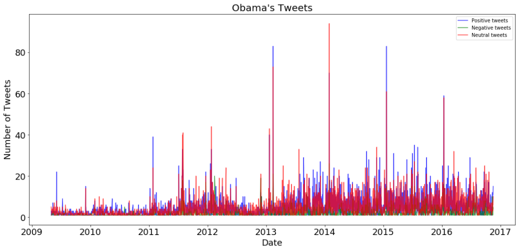
$$Multiplicativemodel : y(t) = trend * seasonality * noise$$

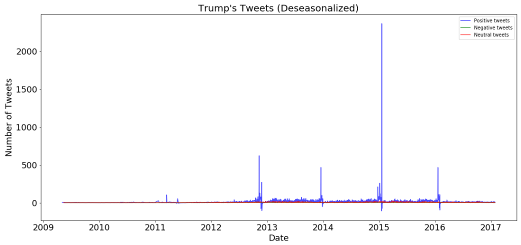
$$Additivemodel : y(t) = trend + seasonality + noise$$

4 Evaluation









Comparing the raw time series plots of both presidents (shown below), we can see that the majority of Obama's tweets were split between being neutral and positive, while Trump's tweets were mostly positive. We considered that neutral tweets may be more indicative of fact-based, rather than opinionated tweets, and infer that Obama may have used his twitter account to relay more objective information than Trump. We also note that Trump tweeted much more often than Obama in the time period that we analyzed, and the quantity did not decrease after he was elected president. This further supports that his tweets may be more reflective of his emotions or opinions on matters, rather than communicating facts to the public.

After deseasonalization, we can see that Trump’s positive tweets increased exponentially in 2015, which was around the time he declared his intent to run for president. When we take a look at Obama’s deseasonalized tweets, he only had a spike of negative tweets in late 2013–this correlates to him attending the funeral of Nelson Mandela, but it does not directly coincide with any events that we found to be negative. However, we do see that Obama has a more balanced ratio of tweets that are positive, negative, and neutral, while Trump tends to sporadically tweet more positively.

Obama Results		
Classifier	Accuracy	F1 Score
Naïve Bayes (Baseline)	0.73	0.77
Naïve Bayes	0.76	0.69
Logistic Regression	**0.91**	**0.87**
Random Forest	0.70	0.65
SVM	0.44	0.20

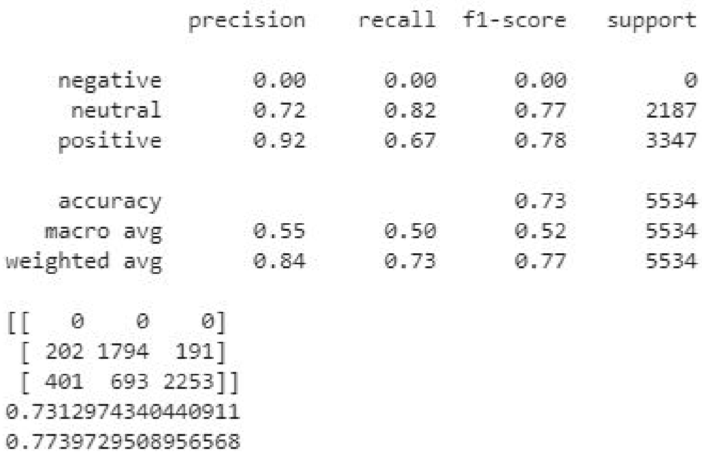
Trump Results		
Classifier	Accuracy	F1 Score
Naïve Bayes (Baseline)	0.59	0.70
Naïve Bayes	0.72	0.74
Logistic Regression	**0.93**	**0.93**
Random Forest	0.79	0.80
SVM	0.54	0.71

We reported all our scores for the models we trained on the two datasets in Table 1. We split our dataset into a 80% training set and 20% testing set and used a 5-fold cross validation before getting our results. The performance of each classifier was estimated by generating a classification report and confusion matrix. Our baseline classifier was the naïve Bayes classifier without any parameter tuning, which gave an accuracy of 0.59 and a F1 score of 0.70 on the Obama dataset. After implementing a grid-search with a 5 cross-fold validation, where the dataset is partitioned into five equal size subsets, we we were able to improve our results up to a 0.72 accuracy level and 0.74 for our F1 score.

For example, using naïve Bayes, our parameters indicated that the best results would be achieved from using a smaller alpha, both unigrams and bigrams, and not using IDF for the transformer resulted in better accuracy and F1 scores:

```
{’bow__ngram_range’: (1, 2), ’clf__alpha’: 0.01, ’tfidf__use_idf’: False}
```

Baseline Naïve Bayes



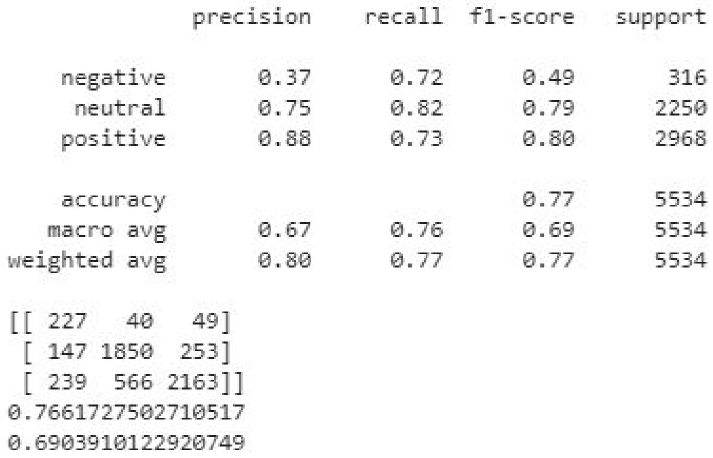
```
              precision    recall  f1-score   support

    negative       0.00      0.00      0.00         0
     neutral       0.72      0.82      0.77      2187
    positive       0.92      0.67      0.78      3347

    accuracy                           0.73      5534
   macro avg       0.55      0.50      0.52      5534
weighted avg       0.84      0.73      0.77      5534

[[   0    0    0]
 [ 202 1794  191]
 [ 401  693 2253]]
0.7312974340440911
0.7739729508956568
```

GridSearch Naïve Bayes



```
              precision    recall  f1-score   support

    negative       0.37      0.72      0.49       316
     neutral       0.75      0.82      0.79      2250
    positive       0.88      0.73      0.80      2968

    accuracy                           0.77      5534
   macro avg       0.67      0.76      0.69      5534
weighted avg       0.80      0.77      0.77      5534

[[ 227   40   49]
 [ 147 1850  253]
 [ 239  566 2163]]
0.7661727502710517
0.6903910122920749
```

Our results showed that logistic regression is best at predicting the sentiment of tweets. This makes sense, as logistic regression often works well with high dimensional data. In both Obama and Trump's datasets, we saw that the accuracy was about 90%, while Trump's dataset did better in the F1 score. However, since the F1 score was relatively high, it meant our model did not just label every tweet as positive, but correctly identified most tweets. While naïve bayes and random forest also gave pretty decent results, we were unable to raise the accuracy of the support vector machine higher than 0.54

5 Conclusion

Social media has played an important role in the political campaigns over the years and with the Democractic and Republic Conventions approaching in 2020, tweets are scrutinized now more than ever. We found similar results that Yaqub et al found in their research–that Trump had more positive tweets and Obama had an equal amount of positive and neutral tweets. This could be due to the nature of the tweets, as Trump tweeted more opinions and emotions, while Obama tweeted more facts. In our research, we found that Trump's tweets tend to be mostly positive before 2015, after which they become mainly neutral, while Obama's tweets maintain their predominantly neutral sentiment throughout this time period. We also noticed that Obama's positive tweets utilized a wider vocabulary than Trump's positive tweets did. In future work, we will explore a more in-depth analysis by utilizing semantic analysis and topic modeling.

References

[1] "Sentiment Analysis 101." KDnuggets, https://www.kdnuggets.com/2015/12/sentiment-analysis-101.html.

[2] "Time Series Analysis in Python - A Comprehensive Guide with Examples – ML ." Machine Learning Plus, 14 Feb. 2019, https://www.machinelearningplus.com/time-series/time-series-analysis-python/.

[3] Alan. "Why Accuracy Alone Is a Bad Measure for Classification Tasks, and What We Can Do about It." Tryolabs Blog, 25 Mar. 2013, https://tryolabs.com/blog/2013/03/25/why-accuracy-alone-bad-measure-classification-tasks-and-what-we-can-do-about-it/.

[4] Čišija, Merima Zunic, Emir Donko, Dzenana. (2018). Collection and Sentiment Analysis of Twitter Data on the Political Atmosphere. 1-5. 10.1109/NEUREL.2018.8586980.

[5] Jr, Sigmundo Preissler. "Seasonality in Python: Additive or Multiplicative Model?" Medium, Medium, 20 Nov. 2018, https://medium.com/@sigmundojr/seasonality-in-python-additive-or-multiplicative-model-d4b9cf1f48a7.

[6] Monsters, Data. “Text Preprocessing in Python: Steps, Tools, and Examples.” Medium, Medium, 15 Oct. 2018, https://medium.com/@datamonsters/text-preprocessing-in-python-steps-tools-and-examples-bf025f872908.

[7] Scott, William. “TF-IDF for Document Ranking from Scratch in Python on Real World Dataset.” Medium, Towards Data Science, 21 May 2019, https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089.

[8] Wahome, Ronald. “This Is How Twitter Sees The World : Sentiment Analysis Part Two.” Medium, Towards Data Science, 9 Sept. 2018, towardsdatascience.com/the-real-world-as-seen-on-twitter-sentiment-analysis-part-two-3ed2670f927d.

[9] Yaqub, Ussama Chun, Soon Atluri, Vijayalakshmi Vaidya, Jaideep. (2017). Analysis of political discourse on twitter in the context of the 2016 US presidential elections. Government Information Quarterly. 34. 10.1016/j.giq.2017.11.001.

[10] Zornoza, Jaime. “Visualisation of Information from Raw Twitter Data-Part 2.” Medium, Towards Data Science, 27 June 2019, https://towardsdatascience.com/visualisation-of-information-from-raw-twitter-data-part-2-11707a65e920.