Sentiment Analysis of Tweets for the 2016 US Presidential Election

Brandon Joyce
Dept. Computer Science
UNC Greensboro
Greensboro, NC 27412, USA
bwjoyce@uncg.edu

Jing Deng
Dept. of Computer Science
UNC Greensboro
Greensboro, NC 27412, USA
jing.deng@uncg.edu

***Abstract*—Twitter is a popular micro-blogging social media platform. For the 2016 US Presidential election, many people expressed their likes or dislikes for a particular presidential candidate. Our work's aim was to calculate the sentiment expressed by these tweets, and then compare this sentiment with polling data to see how much correlation they share. We used a lexicon and Naive Bayes Machine Learning Algorithm to calculate the sentiment of political tweets collected one-hundred days before the election. We used manually labeled tweets as well as automatically labeled tweets based on hashtag content/topic. Our results suggest that Twitter is becoming a more reliable platform in comparison to previous work. By focusing on tweets 43 days before the election (beginning with the first presidential debate), we found a correlation as high as 94% to polling data using a moving average smoothing technique.**

***Index Terms*—Machine Learning, Natural Language Processing, Sentiment Analysis, Social Media, Twitter**

I. Introduction

The US Presidential Election of 2016 was historic for many reasons. The Washington Post called it the "most negative presidential election of our lives" [1]. Many people expressed their feelings for each candidate on different social networks including Twitter, a popular micro-blogging site. Each micro-blog is referred to as a "Tweet" and can be no more than one-hundred and forty characters long. Many tweets also include a label for other Twitter users they are referencing (e.g. @username), and a "hashtag" that usually indicates the topic of the tweet (e.g. #election2016). The growth of Twitter users since the last election suggests that it may have become a more accurate polling tool since the 2012 election. For example, according to Statista, the number of monthly active Twitter users worldwide from 4th quarter 2012 to 4th quarter 2016 grew from 185 million to 328 million [2].

Tweets carry sentiments from their senders and it is important to understand such sentiments. Major events such as presidential elections and users reactions on social networks can be used to understand how users express themselves. Presidential elections are unique in the sense that voters' opinions are usually carefully polled and published. With such data, it is then possible to understand how tweeter sentiments and voter opinions correlate.

Furthermore, twitter users can express their sentiment of a candidate using a hashtag. Since the presidential election is largely dominated by two parties, a hashtag that is used to convey positive sentiment for one candidate might express negative sentiment for the other candidate and vice versa. For our work, we assume that the majority of tweets that express positive "candidate focused" hashtags plan to vote for that candidate on the day the tweet was posted. Likewise, we assume that the majority of tweets that express negative "candidate focused" hashtags plan to vote for the complementary candidate on the day the tweet was posted (note, we are ignoring third-party candidates for this analysis in order to assume that our hashtag sentiment is "binary").

In our approach, we first apply a lexicon sentiment analysis technique as well as a Naive Bayes algorithm to classify tweets. To accomplish the latter, we manually labeled a set of tweets for both Donald Trump and Hillary Clinton. We then compared our results with polling data and with well-known twitter hashtags. Then we used tweets that were automatically labeled positive and negative based on a few candidate specific hashtags we selected. For example, we would expect the hashtag "#imwithher" to express positive sentiment towards Hillary Clinton and possibly negative sentiment towards Donald Trump, and the hashtag "#makeamericagreatagain" to express positive sentiment towards Donald Trump and possibly negative sentiment towards Hillary Clinton.

We will use the next two sections to explain related work and sentiment analysis methodology, including the Naive Bayes Algorithm. Section 4 briefly explains our data collection. Section 5 examines our results. The last section is the conclusion of our work.

II. Related Work

A. Twitter and Polling Data

In [3], O'Connor et al. gathered tweets from the 2012 presidential election that contained the phrase "McCain" or "Obama." They applied a lexicon sentiment analysis technique to the gathered election tweets and compared their results to polling data. They used the subjectivity lexicon from OpinionFinder, which has approximately 1,600 positive words and 1,200 negative words [4]. To compare sentiment scores for tweets to polling data, O'Connor et al. used the ratio of positive to negative tweets for a particular topic (i.e presidential candidate). They achieved a correlation factor as high as 44% using this particular method.

In [5], Jahanbakhsh and Moon used a Naive Bayes Algorithm to classify tweets relating to the 2012 presidential election. They manually labeled 989 tweets to train the classifier. When they compared their results to polling data, they were "mostly in match with [their] Twitter results but with some latency."

B. Sentiment Analysis using Tweets

In [6], Pak and Paroubek used the Naive Bayes Algorithm to classify tweets with emoticons (e.g ":)" would be a positive label and ":(" would be a negative label) They achieved 81% accuracy using two classes (positive and negative), instead of three (positive, negative, and neutral).

In [7], Wang et al. tried to automatically label tweets based on emotion. They used hashtags labels such as "#happy" and "#sorrow" to train a classier to identify tweets that expressed joy or sadness. They achieved an accuracy as high as 65.65%. In [8], Davidov et al. performed a variety of sentiment analyses using tweets. This included using hashtags as labels to train a classier to identify "focused" sentiment. For example, a hashtag that includes an emotion and a target such as "#tmobilesucks" could be used to calculate the sentiment twitter users express toward T-Mobile.

III. Sentiment Analysis Methodology

A. Lexicon

We used the OpinionFinder Lexicon [4]. This lexicon contains roughly 1,600 and 1,200 positive and negative words. As in [3], we did not utilize the weak/strong labels the lexicon provided for each word. We combined this lexicon with the lexicon first used in [9] by Hu et al., since this lexicon accounts for some misspellings, which seem to be frequent on social media. After combining both lists, we checked if any words were labeled as positive and negative. If this was the case, we labeled the word as only positive (there were less than one percent of such words). Then, any duplicate words in the list were removed. We also added a few explicit words to this list and labeled them as negative and removed the words "trump" and "trumpet" for obvious reasons. To calculate a sentiment score for a tweet, we counted the number of positive words and subtracted them from the number of negative words. If the result was negative, we labeled the tweet as negative. If the result was positive, we labeled the tweet as positive. Otherwise, the tweet was labeled as neutral and was not used in our lexicon sentiment analysis [6].

B. Naive Bayes Algorithm

The Naive Bayes Algorithm is based on the following formula

$$P(\text{label}|\text{feature}) = \frac{P(\text{label})P(\text{feature}|\text{label})}{P(\text{feature})}$$

We used two labels: positive and negative. The features are the words in the tweets. Note that we removed stopwords from the tweets, as well as words with a length less than three. We used the National Language Toolkit (NLTK) to implement the Naive Bayes Algorithm [10]. As indicated in the NLTK documentation, this algorithm makes the "naive" assumption that any given word/feature has an independent probability from another word/feature. Thus, the $P(label|tweet)$ for a tweet containing n words can be calculated as:

$$P(\text{label}|\text{tweet}) = \frac{P(\text{label})P(\text{word}_1|\text{label})...P(\text{word}_n|\text{label})}{P(\text{feature})}$$

For the Naive Bayes Algorithm, we labeled 500 negative and 500 positive tweets for both Donald Trump and Hillary Clinton. We labeled tweets from our positive, negative, and neutral tweets calculated during our lexicon analysis. During labeling, we tried to correct any obvious misspellings, as well as isolate key terms from hyperlinks (e.g. "WikiLeaks," "imwithher," etc.). We then used the Naive Bayes Algorithm to classify the tweets we had collected.

However, manually labeling tweets is a very time consuming process. We attempted to automate the process by using "candidate specific" hashtags that we felt expressed a strong sentiment for a particular candidate. We used the hashtags "#imwithher", "#strongertogether", and "#nevertrump" as positive labels for Clinton and negative labels for Trump. Similarly, we used the hashtags "#draintheswamp", "#lockherup", and "#makeamericagreatagain" as positive labels for Trump and negative labels for Clinton. We randomly labeled twenty thousand tweets in total that contained at least one of the aforementioned hashtags (so duplicate tweets were possibly labeled) and the candidates name. Thus, there were five thousand positive and negative tweets for each candidate.

C. Sentiment Scoring

In [3], O'Connor et al. calculated the sentiment score x for a particular day t to be:

$$x_t = \frac{\text{count}_t(\text{pos. tweets} \wedge \text{topic word})}{\text{count}_t(\text{neg. tweets} \wedge \text{topic word})} \quad (1)$$

Where the topic word is either "Donald Trump" or "Hillary Clinton." A similar formula is given by O'Connor et al. [3], except that they essentially counted positive/negative words instead of positive/negative tweets. Thus, a tweet could be labeled as positive and negative.

However, this method does not generate very smooth data. O'Connor et al. used a moving average to smooth their data [3]. In essence, a moving average (MA) uses the average of the past k days to calculate a sentiment score on a particular day t. The formula for this moving average is:

$$MA_t = \frac{1}{k}(x_{t-k+1} + x_{t-k+2} + ... + x_t)$$

This smoothing method helps us to be able to compare our data to opinion polls, since they use similar smoothing techniques.

IV. Data Collection and Polling Data

For this work, we collected tweets from July 31st through November 7th (the day before the election) that contained the keywords "Hillary Clinton" or "Donald Trump." For brevity, those tweets containing "Hillary Clinton" will be called "the Clinton tweets" hereafter, and those containing "Donald Trump" are called "the Trump tweets." The script we used to collect tweets utilized the Twitter Search Engine [11]. The script does not guarantee that every public tweet from the specified date range/query is gathered. The fields for each Tweet include id, username, text, date, etc. Unfortunately, the tweets we collected did not include location data. We collected roughly 3,068,000 tweets mentioning Donald Trump, and roughly 4,603,000 mentioning Hillary Clinton (with some

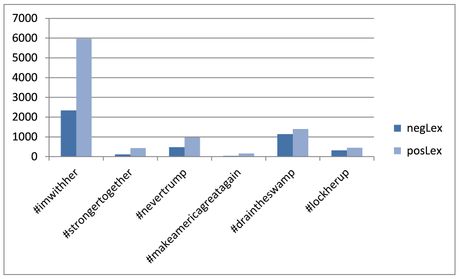

Fig. 1: Lexicon Sentiment for Clinton Hashtags

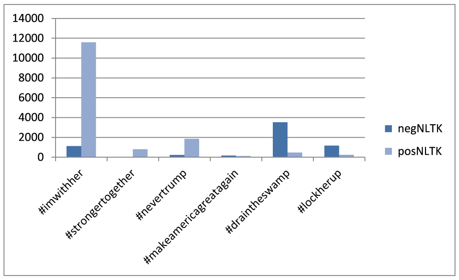

Fig. 2: NLTK Sentiment for Clinton Hashtags

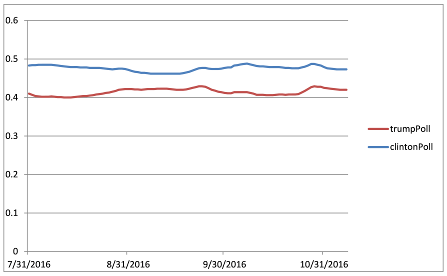

Fig. 5: Polling Data

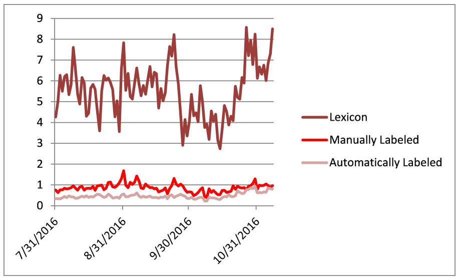

Fig. 6: Trump Tweet Sentiment Scores by Day

possible overlaps) from July 31st-November 7th, 2016. Like O'Connor et al., we used polling data from Pollster.com [12]. This Poll combines 315 polls from 38 pollsters. The last 100 days of this poll are shown in Figure 5.

V. Results

A. Hashtag Sentiment

To help us see how well our Naive Bayes Algorithm was working, we examined tweets that contained popular hashtags. These hashtags were special in that we expected the sentiment of the hashtag to either be positive or negative. For example, we expect the hashtags "imwithher" and "nevertrump" to express positive sentiment for Clinton, but negative sentiment for Trump. However, in Figure 3, these hashtags in the Trump tweets were mostly labeled as positive instead of negative during our lexicon analysis. We can see that in Figure 1 the Clinton tweets with these hashtags were mostly labeled as positive. So we can conclude that our lexicon analysis could not identify positive Clinton sentiment as negative Trump sentiment. However, in Figure 4, the Naive Bayes Algorithm seems to be able to identify positive Clinton hashtags as expressing negative sentiment towards Trump.

It is also interesting to note how the sentiment expressed by the tweets we collected respond to current events. For example, in Figure 7, there is a large drop in Clinton's sentiment score on October 28th, when former FBI Director James Comey announced that new emails had been uncovered in the Clinton investigation [13]. In Figure 6, there is a large drop in Trump's sentiment score on October 7th, when a tape leaked regarding Trump's so-called "Locker Room" conversation [14].

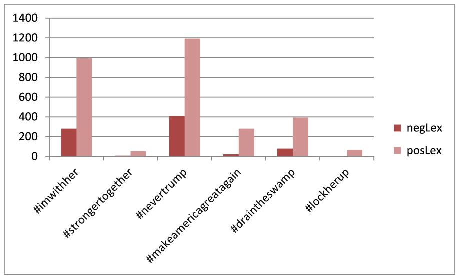

Fig. 3: Lexicon Sentiment for Trump Hashtags

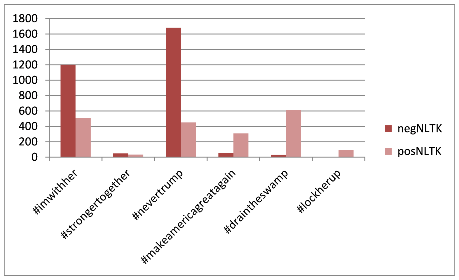

Fig. 4: NLTK Sentiment for Trump Hashtags

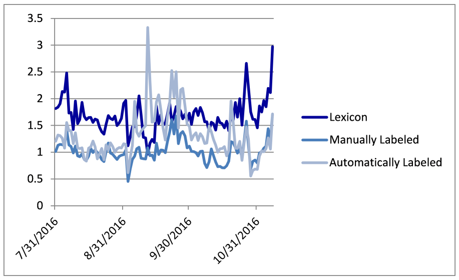

Fig. 7: Clinton Tweet Sentiment Scores by Day

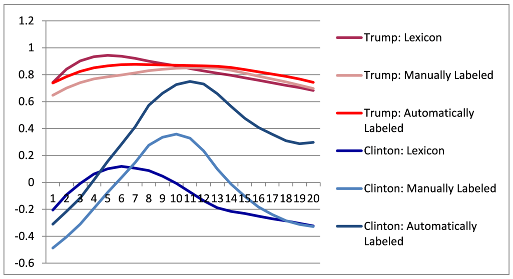

Fig. 8: Correlation Coefficients for Trump and Clinton Tweets 43 days before election using a Moving Average of k days

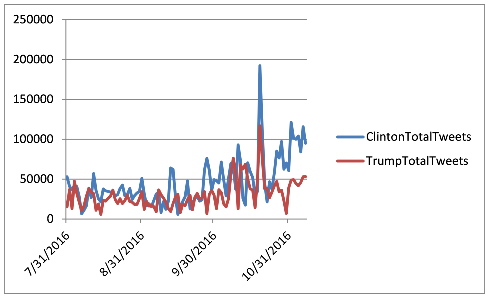

Fig. 9: Tweet Volume Chart

B. Correlation to Opinion Polls

We tested our method for all of the tweets we gathered 100 days before the election. We found a correlation coefficient around 40%-60% for most of our methods (with the exception of the automatically labeled Clinton tweets, which had a negative correlation for $k < 14$). However, 100 days before the election is a "long" time in terms of social media. For example, there were no debates in August, so one can assume that social media was more "quiet" in August than in late September or October. If we focus on the date ranges with more popular events (i.e debates), we should have more tweets (see Figure 9) and hopefully more accurate results.

If we look at all the tweets beginning on the date of the first debate through the day before the election (43 days total), we get very different results. For the Clinton tweets, our Naive Bayes Algorithm had a surprising negative correlation of 48.77% using a window of $k = 1$ and a positive correlation of 35.85% using a window of $k = 10$ (See Figure 8). For the Trump tweets, our Naive Bayes Algorithm had a positive correlation as high as 85.19% using a window of $k = 12$ (See Figure 8). For our lexicon analysis, the Trump tweets had a correlation of 94.42% using a window of $k = 5$, and the Clinton tweets had a negative correlation of 20.53% using a window of $k = 1$. Thus, the Trump tweets appear to correlate very well to the polling data we used. The automatically labeled tweets using the hashtags we selected seem to perform better than the manually labeled tweets. The Trump tweets had a correlation of 87.68% using a window of $k = 7$. The Clinton tweets had a correlation of 74.98% using a window of $k = 11$ and this correlation was not negative for $k > 3$ unlike the other two methods. In general, a window of 11-14 seems to produce better results.

VI. Conclusions

In this work, we have investigated the accuracy of a lexicon sentiment analysis and a machine learning sentiment analysis when the results are compared to polling data. Even though the Naive Bayes Machine Learning Algorithm seemed to do well identifying sentiment associated with particular hashtags, it did not outperform the lexicon analysis as anticipated when compared with Trump polling data. However, the automatically labeled tweets outperformed the manually labeled tweets for both candidates and have better accuracy when compared to the Clinton lexicon analysis. Thus, the automatic method saves man-hours, improves accuracy, and removes any potential bias that could occur when the tweets are being manually labeled. The very high correlation coefficient with our trump tweets suggests that Twitter is becoming a larger and more diverse platform that is beginning to rival sophisticated polling techniques. Perhaps in the future, social media polls will become more incorporated into polling schemes.

Acknowledgment

This research was funded by the National Science Foundation, Award# CCF-1320428 under the Research Experience for Undergraduates Program.

References

[1] A. Blake, "Welcome to the next, most negative presidential election of our lives," http://wapo.st/2a9uWr8?tid=ss_mail, Washington Post, accessed: 2017-05-03.

[2] Statista, "Number of monthly active twitter users worldwide from 1st quarter 2010 to 1st quarter 2017 (in millions)," https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/, accessed: 2017-05-03.

[3] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From tweets to polls: Linking text sentiment to public opinion time series." *ICWSM*, vol. 11, no. 122-129, pp. 1–2, 2010.

[4] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of the conference on human language technology and empirical methods in natural language processing*. Association for Computational Linguistics, 2005, pp. 347–354.

[5] K. Jahanbakhsh and Y. Moon, "The predictive power of social media: On the predictability of us presidential elections using twitter," *arXiv preprint arXiv:1407.0622*, 2014.

[6] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining." in *LREc*, vol. 10, no. 2010, 2010.

[7] W. Wang, L. Chen, K. Thirunarayan, and A. P. Sheth, "Harnessing twitter" big data" for automatic emotion identification," in *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*. IEEE, 2012, pp. 587–592.

[8] D. Davidov, O. Tsur, and A. Rappoport, "Enhanced sentiment learning using twitter hashtags and smileys," in *Proceedings of the 23rd international conference on computational linguistics: posters*. Association for Computational Linguistics, 2010, pp. 241–249.

[9] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 168–177.

[10] E. L. Bird, Steven and E. Klein, "Natural language processing with python," OReilly Media Inc., 2009.

[11] Jefferson-Henrique, "Getoldtweets," https://github.com/Jefferson-Henrique/GetOldTweets-java, accessed: 2017-05-03.

[12] Pollster, "2016 presidential election," http://elections.huffingtonpost.com/pollster/2016-general-election-trump-vs-clinton, accessed: 2017-05-03.

[13] CBS, "A james comey timeline," http://www.cbsnews.com/news/a-james-comey-timeline/, accessed: 2017-05-03.

[14] AOL, "Election timeline," https://www.aol.com/2016-election/timeline/, accessed: 2017-05-03.