Twitter Sentiment Analysis for US election 2020

Anshuman Singh, Yixin Fan, Dalila Islas Sanchez, Divyansh Oze
School of Physics and Astronomy
University of Nottingham

***Abstract*—Sentiment Analysis is performed on the Twitter data to classify the polarity of the user's tweets, which can be beneficial to companies, organisations or individuals to use insights in more profound actions. Twitter contains tremendous amounts of data on the platform, most unstructured, making it difficult to process data using traditional methods. This paper aims to use Big Data Techniques to process tweets as data from the US Election 2020 and carry out a model to classify the tweets into respective categories of emotion and subjectivity. However, the result shows that both candidates had a comparable number of positive and negative tweets. Donald Trump had more negative tweets state-wise for the US, with some states having a margin of 2x rates. This paper provides a brief insight on the sentiment analysis for Twitter data, the relationship between the results observed from classification and the conclusion.**

Twitter, Big Data Processing, Natural Language Processing, Sentiment Analysis.

I. Introduction

Twitter is a popular social communication tool where users can post messages. These tweets could express users' ideas on different topics, such as product reviews. Therefore, sentiment analysis is introduced to analyse tweets' polarity, which helps to understand people's attitudes toward a particular topic.

However, since the tweets data is unstructured, one of the challenges of Twitter sentiment analysis is how to deal with these data effectively. Besides, as the data size has become increasingly large, the big data techniques has shown its great advantages when processing a large amount of unstructured data compared to the traditional way. This paper will focus on the sentiment analysis of the US 2020 election using big data techniques.

This section introduces the background of sentiment analysis on the US 2020 election based on Twitter data. Section 2 will include the detailed problem description and the literature review of the previous work, followed by section 4 regarding the result discussion. Finally, this paper will conclude and discuss some future work.

II. Literature Review

Sentiment analysis is an interdisciplinary task which includes data scrabbling, natural language processing, and machine learning. This section will focus on the relative work about tweets sentiment analysis that has been done by other researchers, as well as the corresponding big data techniques.

A. Sentiment Analysis task

The sentiment analysis divides into following three steps [1]:

- Subjectivity Classification: This is the first step that takes the tweets that express users' opinions.
- Sentiment Classification: Classify the opinionated tweets by different polarities, such as positive, negative and neutral.
- Complimentary Tasks: This includes Object/Holder Extraction, which focuses on discovering a tweets' source, and Object/Feature Extraction, which is the target entity.

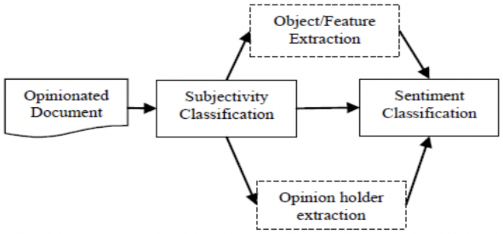

Object/Feature Extraction
Opinionated Document
Subjectivity Classification
Sentiment Classification
Opinion holder extraction

Fig. 1: Sentiment Analysis Task [1]

B. Approaches for Sentiment Classification

The approaches for Twitter sentiment analysis divides into two types:

Machine Learning Approaches After collecting the sample datasets, the next step is to train a classifier on the data. Finally, this classifier implements to make predictions. It consists of unsupervised learning and supervised learning.

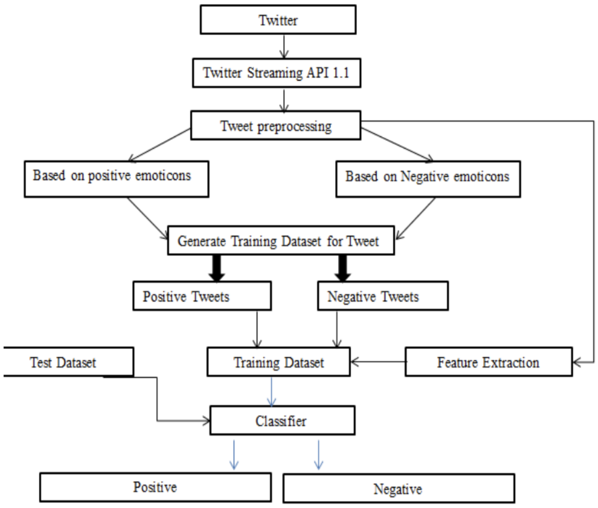


Fig. 2: Sentiment Classification Based On Emoticons

Specifically, unsupervised learning works with unlabeled datasets(the target will not be provided) and hence it rely on the cluster, whereas supervised learning are based on the dataset with label. Besides, classifier selection is also a significant step. There are three commonly used Machine Learning techniques:

- Naive Bayes: this is a simple probabilistic classifier based on the Bayes' Theorem. This classifier could be trained directly by the NLTK library in Python. [2].
- Support Vector Machine: It has shown significant advantages in dealing with large feature space, and research concludes a much higher accuracy than the traditional algorithms. However, it still has a black box problem, which makes it challenging for people to figure out which word is more important than the others. [2], [3].
- Max Entropy: This classifier always aims to maximise the entropy by estimating the conditional distribution of the class label. Soni has researched the Naive Bayes classifier and the Maximum Entropy classifier. The result shows that the Maximum Entropy classifier performs better than others in predicting the sentiments with an accuracy of 74% [4].

Lexicon-Based Approaches These Approaches are widely used in text classification. Its core idea is to determine the emotion in text data using the sentiment dictionary which contains opinionated words and then match with data text. Hence it relies on the words that is included in the sentiment dictionary.

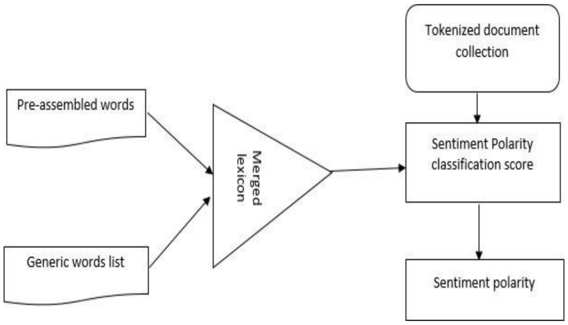


Fig. 3: Lexicon-Based Model

- Dictionary-based: This method is based on manually collected and annotated terms, which grows by searching for synonyms and antonyms in the dictionary, such as WordNet.
- Corpus-Based: Corpus based approaches have domain specific dictionaries. They are created on a set of seed terms which further increases through the relevant word search by use of semantic or statistical methods.

C. Big data techniques

As mentioned in the above section, the traditional data processing methods cannot handle a large amount of unstructured Twitter data effectively. Therefore, many researchers have introduced big data techniques to solve the problem. Trupthi et al. have carried out research which has used a system to train the model with the help of MapReduce and Hadoop. They also used a Naïve Bayes classifier, Time-Variant Analytics and the Continuous-learning System, and the result costs less and analysis is done by real-time tweets data [5]. Rodrigues et al. also introduced a new big data technique based on Hadoop Framework and a new classifier, Hybrid Lexicon-Naive Bayesian Classifier. The result has shown great improvement with the accuracy of 82% and achieves 93% of improvement in time cost [6].

III. Methodology

This section describe the methodology, including data description, data preprocessing, features and approaches.

A. Data Description

The dataset we used for sentiment analysis is based on the 2020 Election in the United States of America. We have made use of two csv files that were scraped off from Twitter using the Twitter API by the keywords of Donald Trump and Joe Biden. The tweets date for the two data frames ranged from 15/10/2020 to 04/11/2020.

The columns in the dataset have all sorts of information, from the date and time of tweet creation ('created_at') to unique personal ('user_id', 'user_name') and geographical details ('city',' country',' lat', 'long') about the user who has put the tweet online.

For the complete sentiment analysis, we need full-length tweets as inputs for our model to perform actions in completing the prediction. We have used the 'tweet' column, which contains the full tweet text tweeted by the user online.

To further add to this, there is another exciting column, 'user_description', a self-description by the tweet creator, which can be used as a feature to study an individual's personality for a more complex model in sentiment analysis.

We made use of PySpark to process the data as inputs, since it made the process simpler to replace delimiters and enabled us to read multiple lines efficiently.

B. Data Preprocessing

Twitter as a platform has people commenting on everything possible going around in the world, from something catastrophic to euphoric to general tweets, which are just personal updates from their daily lives. Then, the tweets naturally contain different users' opinions expressed in erratic ways. It is crucial to preprocess the dataset right before implementing any models to avoid wasting computational memory, time and objects in the data that can mislead the classification model for its predictions on the tweets. For the significant chunk of preprocessing the twitter-election-2020 dataset, The NLTK module is for Natural Language Processing, which has helped us immensely process the data and implement built-in functions from the NLTK toolkit to make the prediction process smoother and more efficient with the insights achieved from the dataset.

1) Data Cleaning

The raw data can negatively affect our model for processing the inputs and predicting the polarity classes with imbalanced or unstructured data. The pre-processing section has worked on these points for what got scrapped off from the tweets:

- It removes all the URLs (e.g. zyx.com) in the full-length tweet text.
- It removes special characters that are of no use to model in predicting the sentiment.
- It removes non-English words and punctuations and converts all words into a lower case for simplicity.

There has also been the usage of Regex functions to replace words like the United States, America, and United States of America to 'US' as it can mislead our model to classify these tweets as from different countries. The parallel goal of any machine learning or big data project is to save computational time and memory while performing the task. To make our process more efficient, we make use of more built-in NLTK tools to further prepare our data for predictions:

Tokenize We use the in-built tokenise function to break the sentences into small individual fragments. It further moves the punctuation and lands the tokenised words to stemming and lemmatisation. E.g. this is a sample:- "this", "is", "a", "sample"

Stemming The tokenised words would pass through a stemmer, which chops off the end of a word in a crude fashion, relying upon it to make sense. E.g. changing, changed, change- chang.

Lemmatize Unlike stemming, lemmatise is not so crude with its chopping. Lemmatise instead attempts to return a word with the use of a dictionary. E.g. Changing, changed, change: change.

C. Features

The features for the project task are the text inputs as tweets that get cleaned during the pre-processing and data preparation section. They are further used to make the classification task for sentiments easier based on these texts as features.

D. Approaches

The lexicon-based approach is used to perform analysis on Twitter data. As the lexicon-based approach heavily depends on the polarity of sentences and words, we have used the TextBlob function to calculate polarities for each passing tweet in our dataset as texts. The polarities further on can be classified into three different sentiments, “negative”, “neutral”, and “positive.

TextBlob:
As a lexicon-based sentiment analyser, given its knowledge of word and weight dictionaries, it helps us determine the following things with ease for our tweets:

- The TextBlob can be used to calculate the subjectivity of the tweet or sentence. It identifies if the text is of a subjective or objective class.
- It calculates the polarity for a tweet, or the sentence passed, to further give out a sentiment between negative, neutral and positive.

User-Defined Functions:
The UDFs, commonly known as User-Defined Functions, are the functions defined by the user in the environment. We have made such UDFs to benefit from the fact that they can be re-used on multiple DataFrames and SQL just after registering once.

Following are the UDFs that have helped the Lexicon Approach to work for predictions:

1) **get_polarity:** The TextBlob returns the polarity of the respective text passed to the function.
2) **get_subjectivity:** The TextBlob returns the subjectivity of the respective text passed to the function.
3) **classify(polarity):** Based on the polarity achieved from the text input passed to the get_polarity function, classify UDF uses that polarity score to place it under three sentiments, i.e. ‘negative’, ‘neutral’, and ‘positive’.

Once the tweets are thoroughly cleaned and passed through NLTK in-built functions to prepare them to act as features for our model, the get_polarity function uses TextBlob to calculate polarity scores on the respective tweets. The project’s primary goal is to analyse the Twitter data and find correlations or patterns between them, using the polarity scores we call the classification function to order out different sentiments for tweets passed. We will further use these classification results to derive patterns and conclusions.

IV. Experiment and Result

This section focuses on the data visualization and results, which will help us investigate our data and summarize their main features.

A. Exploratory Data Analysis

The EDA, also commonly known as the Exploratory Data Analysis, is used to analyse the data. There are numerous methods and techniques to do EDA with the help of spark to maximise our insight into the predictions and dataset and uncover some underlying structure to the results achieved at the end. The data plots displayed below are compared to find a relationship between the number of likes on tweets for both the candidates, the polarity of the sentiment concerning each state in the US, and the count for tweets based on different states for each candidate.

The data plots shown below are relationships and results from the classification task plotted with the count of users’ tweets and the respective candidate.

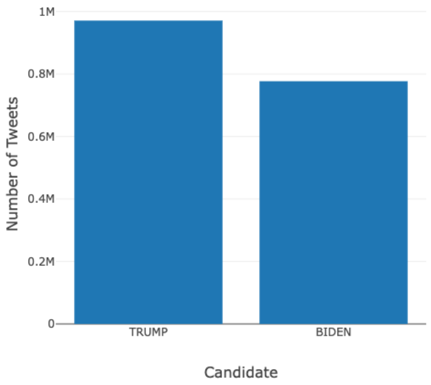


Fig. 4: Number of Tweets for each Candidates

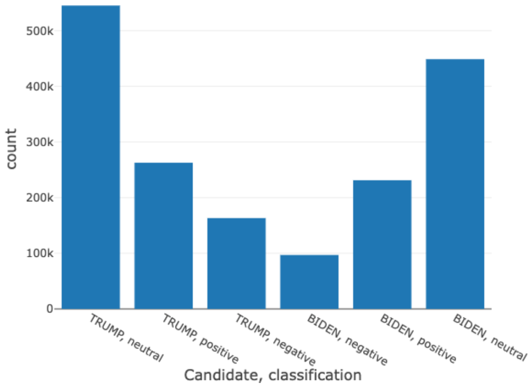


Fig. 5: Summary of Sentiment by Candidates

We use the US election dataset to analyse and predict the sentiments of people's tweets living in different states in the United States of America. Further on, more plots were displayed to draw out a compare the "positive"," neutral", and" negative" tweet sentiments with their tweet count to specific states in the US for both candidates.

Sentiment based results for states and candidates:

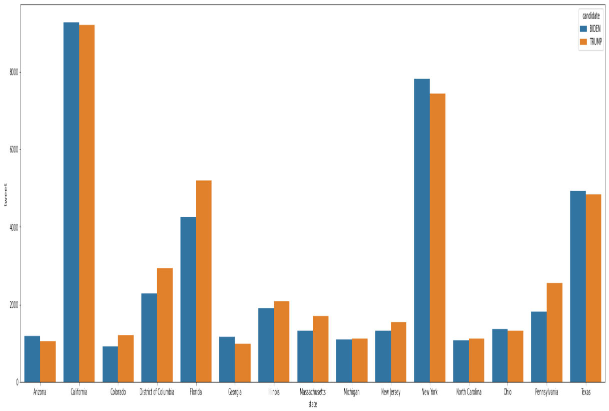


Fig. 6: Positive Sentiment of Two Candidates Comparison

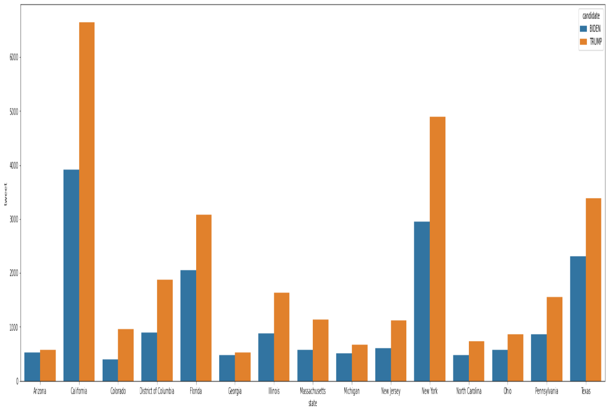


Fig. 7: Negative Sentiment of Two Candidates Comparison

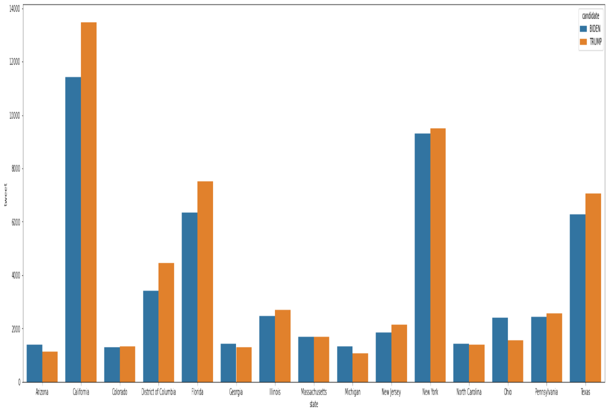
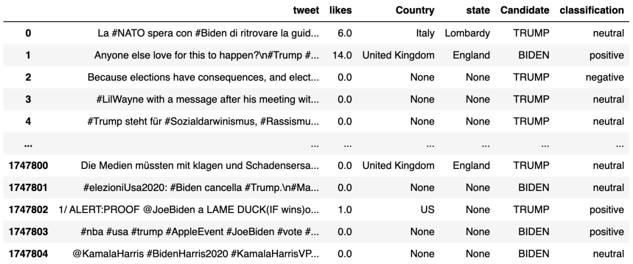


Fig. 8: Neutral Sentiment of Two Candidates Comparison

The following snippet is the final DataFrame with results from the classification user-defined function on the tweets.



	tweet	likes	Country	state	Candidate	classification
0	La #NATO spera con #Biden di ritrovare la guid...	6.0	Italy	Lombardy	TRUMP	neutral
1	Anyone else love for this to happen?\n#Trump #...	14.0	United Kingdom	England	BIDEN	positive
2	Because elections have consequences, and elect...	0.0	None	None	TRUMP	negative
3	#LilWayne with a message after his meeting wit...	0.0	None	None	TRUMP	neutral
4	#Trump steht für #Sozialdarwinismus, #Rassismu...	0.0	None	None	TRUMP	neutral
...	...	...	...	...	...	...
1747800	Die Medien müssten mit klagen und Schadensersa...	0.0	United Kingdom	England	TRUMP	neutral
1747801	#elezioniUsa2020: #Biden cancella #Trump.\n#Ma...	0.0	None	None	BIDEN	neutral
1747802	1/ ALERT:PROOF @JoeBiden a LAME DUCK(IF wins)o...	1.0	US	None	TRUMP	positive
1747803	#nba #usa #trump #AppleEvent #JoeBiden #vote #...	0.0	None	None	BIDEN	positive
1747804	@KamalaHarris #BidenHarris2020 #KamalaHarrisVP...	0.0	None	None	BIDEN	neutral

Fig. 9: Classification of tokenized tweets

The Data Frame has the following columns in it: indexed tweets, tweets, likes on the separate tweet, the country from where the tweet is placed, the state from where the tweet is placed, and the candidate the tweet targeted, with our predicted classification value for the sentiment of the tweet.

V. Conclusion

While moving ahead with the Twitter sentiment analysis with the lexicon approach, it often tends to fail at grabbing the context of the sentences, or if someone were to use a complex, advanced model to understand the personalities of people using the tweets.

Overall, TextBlob provides us with a good enough accuracy on the Twitter responses for polarity, which we used to classify our sentiments. The reason for this is the size of the tweets which apparently do not have too much content, to begin with. This makes it more efficient to use a lexicon approach which works flawless in these conditions.

According to the final results and graphs observed for the US Election 2020, the analysis states that candidate ‘Donald Trump’ had better views on Twitter than ‘Joe Biden’ when considering the whole world. However, compared to the result within the US and its states, Trump had more negative tweets in his name, which correlates with the real-world results.

References

[1] V. A. Kharde and P. S. Sonawane, “Sentiment Analysis of Twitter Data: A Survey of Techniques,” *International Journal of Computer Applications*, vol. 139, no. 11, pp. 5–15, Apr. 2016, arXiv: 1601.06971. [Online]. Available: http://arxiv.org/abs/1601.06971

[2] S. Bhuta, A. Doshi, U. Doshi, and M. Narvekar, “A review of techniques for sentiment analysis Of Twitter data,” in *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, Feb. 2014, pp. 583–591.

[3] X. Ding, B. Liu, and P. S. Yu, “A holistic lexicon-based approach to opinion mining,” in *Proceedings of the 2008 international conference on web search and data mining*, 2008, pp. 231–240.

[4] A. K. Soni, “Multi-lingual sentiment analysis of twitter data by using classification algorithms,” in *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2017, pp. 1–5.

[5] M. Trupthi, S. Pabboju, and G. Narasimha, “Sentiment Analysis on Twitter Using Streaming API,” in *2017 IEEE 7th International Advance Computing Conference (IACC)*, Jan. 2017, pp. 915–919, iSSN: 2473-3571.

[6] A. P. Rodrigues and N. N. Chiplunkar, “A new big data approach for topic classification and sentiment analysis of Twitter data,” *Evolutionary Intelligence*, May 2019. [Online]. Available: https://doi.org/10.1007/s12065-019-00236-3