Twitter Based Sentiment Analysis of Each Presidential Candidate Using Long Short-Term Memory

Dhruval Shah *shah042@cougars.csusm.edu*
Computer Science and Information Systems,
California State University San Marcos
San Marcos, CA, 92096, USA

Yanyan Li *yali@csusm.edu*
Computer Science and Information Systems,
California State University San Marcos
San Marcos, CA, 92096, USA

Ahmad Hadaegh *ahadaegh@csusm.edu*
Computer Science and Information Systems,
California State University San Marcos
San Marcos, CA, 92096, USA

Abstract

In the era of technology and internet, people use online social media services like Twitter, Instagram, Facebook, Reddit, etc. to express their emotions. The idea behind this paper is to understand people's emotion on Twitter and their opinion towards Presidential Election 2020. We collected 1.2 million tweets in total with keyword like *"RealDonaldTrump"*, *"JoeBiden", "Election2020" and* other election related keywords using Twitter API and then processed them with natural language processing toolkit. A Bidirectional Long Short-Term Memory (BiLSTM) model has been trained and we have achieved 93.45% accuracy on our test dataset. We then used our trained model to perform sentiment analysis on the rest of our dataset. With the sentiment analysis results and comparison with 2016 Presidential Election, we have made predictions on who could win the US Presidential Election in 2020 with pre-election twitter data. We have also analyzed the impact of COVID-19 on people's sentiment about the election.

Keywords: Sentiment Analysis, LSTM, Deep Learning, Natural Language Processing, Data Mining.

1. INTRODUCTION

The US presidential election is one of the most important events occurring in 2020 in the United States. Political candidates' tweet about their campaign and their views on various topics and so do the daily twitter users. Due to the advancements in technology, many people have started using platforms like Twitter and Reddit to discuss and share their views on the news and hot topics around the globe. Everyday users post millions of tweets and express their emotions or views on the platform [1]. Performing data analysis on the tweets can be used by the ruling party and opposition party to decide their election strategy. As we are collecting real-time tweets, we can find the live response of public towards the election. Thus, collecting tweets and performing sentimental analysis on them would give us the predictions of who could win the US General Election 2020. In this paper we discuss our approach of using tweets related to US Presidential Elections in 2016 and 2020 and perform sentimental analysis on both. We then try to find the comparisons between the two and make predictions for 2020 Election based on sentimental analysis.

There are many techniques available to do sentiment analysis. For this work, we are going to use Recurrent Neural Network based Long-short Term Memory (LSTM) algorithm. To better understand how LSTM works and how we can use it for our twitter data, we referenced Tang Y. and Liu J.'s work on "Gated Recurrent Units for Airline Sentiment Analysis of Twitter Data" [2], After that we trained LSTM model using labelled tweets for the prediction and sentiment analysis. In section 4, we will discuss about this approach in more detail under Methodology. Section 2 introduces the background of the US election and Twitter where we will discuss the first use of twitter for election campaigning and its impacts on results. In section 3, we will discuss prior work that is emoji-based sentimental analysis on US Elections and why it is not accurate. In section 4, we will discuss about the implementation and methodology of our work in detail. We labelled the data by using VADER API and got 93.45% accuracy on our test data. In this section, we will discuss about this in more detail. In section 5, we will do analysis of our resulted data and subsequently, we will analyze the results of the 2016's and 2020's election. We will take reference of Donna Ladkin's work [3] to understand why Clinton lost the election even though she had more positive sentiment than Trump in 2016. In section 6, we will conclude why our work is more accurate than the previous works and predict the chances of who wins the presidential candidate in 2020.

2. BACKGROUND

Social media is a platform where people express their views and opinions on various issues. Twitter currently receives around 500 million tweets every day (As per the data from August 2013) in which people mostly share their opinions about trending news or headlines [4]. Political parties hire experts who do sentiment analysis of these tweets and help them to decide their strategies for the election campaign. One way to predict the possibility of a candidate winning the election is by doing the sentiment analysis on the tweets.

In 2008, Barack Obama used Twitter to promote his candidacy for the presidential elections [5]. Obama's campaign employed 100 individuals to run his digital presence. This made a huge impact on the election results. Since then, before each election, parties hire research associates to do public sentiment analysis to conduct election manifesto. Through this way, they can understand what their voters want from the next president. People from all over the world tweet about the presidential candidates and the US elections in order to reflect their respective views. Democratic party's presidential candidate Joe Biden and Republican party's presidential candidate Donald Trump were the top ten political trending twitter topics on Twitter on August 12, 2020.

3. RELATED WORK

English is the highest spoken language in the world. Most research efforts are done in English language for sentiment analysis. In 2009, A. Tumasjan, T. O Sprenger, P. G Sandner and I. M Welpe did prediction with Twitter in 2009 and found that 50% users tweeted only one time in their corpus and rest 50% made 90% of the tweets [6]. Their data set was of approx. 100K tweets which is too ambiguous to predict for the large population of a country.

In the paper "A novel classification approach based on Naïve Bayes for Twitter sentiment analysis", they used positive and negative words file to predict the sentiment of a sentence with Naïve Bayes classification. But since the number of positive and negative words are not always equal, there are chances of getting Biased result. [7] To address this issue, they proposed two ways. First one is by counting the number of positive and negative words in calculating weights while seconds approach identifies significant words to predict the class of test document. By this way they achieved 85.33 % accuracy. While it's almost impossible to have all the positive words in positive words file and negative words in negative words file, applying this model to our 10 M tweets dataset may give less accuracy and there are chances of getting wrong sentiment for many tweets.

Gautam, G., & Yadav, D proposed set of techniques of Machine Learning with Semantic analysis. [8] They did performance comparison of Naïve Bayes, Maximum Entropy, Support Vector Machine and Semantic Analysis (WordNet). And they got accuracy ranging between 88.2% to

89.9% accuracy. The accuracy is quite good if we use their approach but this approach will fail to predict correct sentiment sometimes as the word order is not preserved.

Sharma, P. & Moh, [9] did prediction of Indian Election Using Sentiment Analysis on Hindi Twitter. For this work, they used three different algorithms which are Naïve Bayes, SVM and Dictionary Based. They got 62.1%, 78.4% and 34% accuracy respectively.

B. Joyce and J. Deng gathered tweets from 2016's presidential election based on specific words like Hillary Clinton and Donald Trump [10]. They have used Twitter's streaming API for tweets collections. The keywords they included were ***democratic*** and ***republican*** and the full name of the top political candidates like **Donald Trump**. Since they used Twitter's official tweets streaming API, they collected around 79 million tweets by two month's 10,000 unique users. Then they filtered 1.9 million tweets that contain emojis. After deleting retweets, the data set further depleted to 783K tweets. With the *Emojis Sentiment Ranking*, they determine each tweet's emotional orientation. For that, they extracted the emoji from the sentence and then converted it into Unicode. Then applied regex on Unicode and found score from the customize list of 522 unique emoji characters. This score will determine emotional orientation or sentiment of the sentence. To build a sentiment classifier, they used a multinomial Naïve Bayes classifier.

Since not every sentence contains an emoji, only judging the result based on emojis will not be accurate. There may be some tweets where the user has added an emoji as a sarcastic tone. Also, word ordering matters while doing sentiment analysis. For example, one sentence is *I have to read this book* and another sentence is *I have this book to read.* Both examples have different meaning. This simple example is one of the examples where we can say word ordering matters while doing sentiment analysis.

To overcome the above problem, in our work we developed a model that will try to analyze the sentiment based on word number. Deep learning approaches like Long Short-Term Memory (LSTM) is one of the Recurrent Neural Network approaches where model is trained with the word's ordering whereas in the already existing work, they only considered emojis while doing the analysis. Due to the above-mentioned reasons, LSTM seems to be the perfect fit for doing sentiment analysis. So, we will be using LSTM model for our analysis.

4. METHODOLOGY

This section discusses about our methodology. To start with, we downloaded Twitter's 55,000 prelabelled tweets. For training, we will used random 40K tweets and the rest of 15K tweets for testing purposes. But these tweets were not sufficient to train the ML model. As the part of next process, we will collect tweets data of Joe Biden and Donald Trump. For this, we used following keywords: "JoeBiden", "DonaldTrump", "BidenHarris", "US Election 2020", "TrumpPence". We collected around 1.25M tweets for the period of October 2019 to July 2020. Figure 1 shows our proposed system workflow.

4.1. System Overview

With the tweets collected, we did data cleaning and data preprocessing. In Data Preprocessing, we check if sentence language is English or not. If the language is different then we remove those sentences from our dataset. In data cleaning, we remove unwanted texts, URLs, language slangs, emojis, hashtags, mentions, and retweets. After data preprocessing and data cleaning, we were left with almost 1.2M tweets. We randomly took 55K tweets from this dataset. We labelled these tweets using VADER sentiment API [11]. Once these 55K tweets were labelled, we divided 40K tweets as training and rest 15K tweets for testing to make training dataset big enough for model training. We passed these 80K tweets data to train our LSTM model. Before passing tweets to the model, we implemented embedding layer. We used Word2Vc model which initializes random weights and learns to embed all words in the dataset. Once embedding is done, we use LSTM model for RNN layers. To prevent our model from overfitting, we used dropout techniques. It drops irrelevant information from the network as they don't contribute to the process of enhancing our model accuracy. We have used dense layer in this model which connects every input with every output using weights. We have used Relu activation function as it helps complex relationship in the data to be captured by the model. Relu is more efficient than Tanh or Sigmoid activation function because Relu activates certain number of neurons.

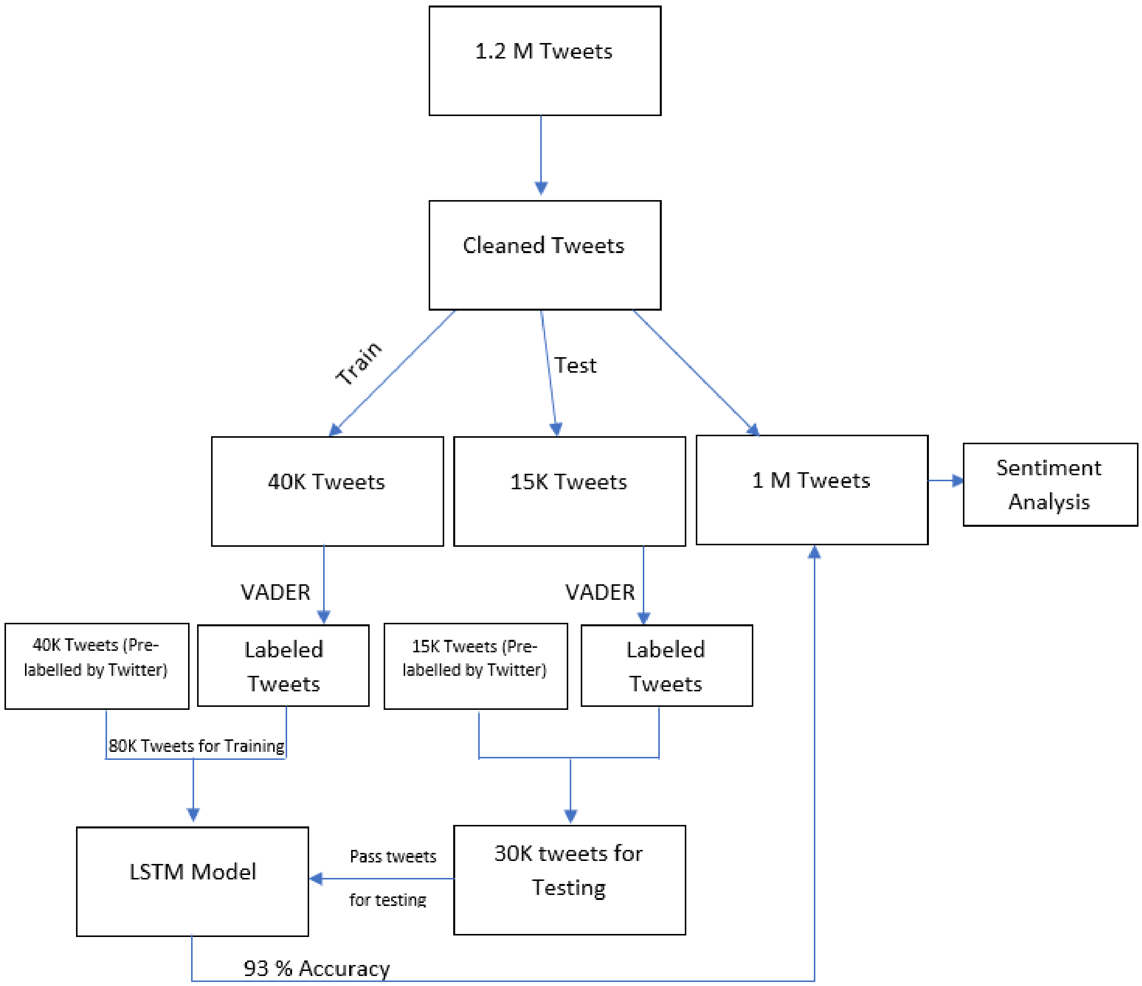

FIGURE 1: Project process workflow.

To test our model, we passed the rest 15K pre labelled tweets from twitter and 15K labelled tweets from VADER API. We got 93.45 % test accuracy which is better than previous works [10].

We passed rest of tweets to this model for labelling. Once these tweets were labelled, we counted the number of positive and negative sentiment tweets. We also passed 2016's tweets data and labeled them. Once the data from both the elections was labelled, we counted number of positive and negative tweets in these to do the analysis.

We have divided the process into 4 major sections below. Data Gathering, Data Processing, Model Training and Testing on test data. All of these steps are very crucial for training the model. The model must be fed with correctly labeled data in order to generate the most accurate results. To give a broader overview, here is the overall process. Using Get Old Tweets (GOT3) API [12] we will collect the tweets. The raw data might be very informal and unstructured. It may contain hyperlinks, emojis, mentions, and retweets. To convert it into trainable data, data cleaning is required. Data cleaning will be performed on tweets that were collected from October 1, 2019 to July 31, 2020 and also October 1, 2015 to November 7, 2016. During the data processing, hashtags, mentions, emojis, links (photo, video, or gifs) and retweets get removed. After that, the data is divided into 80-20 % for training and testing respectively. Once we have the data ready, we use it to feed into the model to train and then test its accuracy with the test data set.

4.2. Data Gathering

Initially, we start collecting tweets using Tweepy streaming API. Since February 1, 2018, standard users cannot access past tweets for more than 7 days. So, we used GOT3 API (Get Old Tweets). GOT3 is a GitHub repository which uses URLlib for fetching tweets from Twitter's advance search. Like this, we collected tweets from October 1, 2019 to July 31, 2020, and October 1, 2015 to November 7, 2016. We randomly selected 1.25M tweets from this time period and their geo-graphical location as United States. By doing data preprocessing, we checked each tweet and filtered out non-English tweets. For this, we used Python's Langdetect library. Then we did data cleaning. In this, we filtered out tweets which only contained emojis, images, gifs, or video. Of 1,250,000 tweets, we obtained 1,200,718 tweets that were containing tweets and may contain images, video, or gifs.

For 2020's data, we used keywords such as "US Elections" with the twitter usernames "realDonaldTrump" and "JoeBiden". Then we collected 2016's tweets for presidential candidates Donald Trump and Hillary Clinton using the same GOT3 API. We collected tweets containing keywords like US Elections, realDonaldTrump, HillaryClinton. Some of the keywords and hashtags we used for data gathering are US Elections, US Elections 2020, TrumpPence, republican election, presidential elections, election 2020, donald trump for 2020, BidenHarris. And we also collected election related tweets from the twitter handles JoeBiden, POTUS, DonaldTrump, KamalaHarris, Mike_Pence. In the data gathering, we only collected tweets from United States. We did this by filtering tweets by its location (in our case it's United States).

Example 1: @JoeBiden VP Biden I wish you wouldn't debate Trump unless he: Debate #1 Trump shows taxes Debate #2 Tells Putin stop killing our soldiers Debate #3 Put sanctions on Putin for 2016 election hacks & I wish you'd publicly challenge him with this. #Biden/Rice 2020

Example 2: Best @realDonaldTrump political ad I've seen yet. @TheDemocrats. will attempt just about anything to impact the 2020 election. Pure evil. cc @Scavino45 @parscale

Example1 and Example 2 are the sample tweets from the training dataset and the words in bold are those words which will would cause problem while training model. So, we need to remove these words, special characters, URLs and images from the tweets. We remove them during Data Processing.

4.3. Data Processing

As the language used on Twitter is informal and unstructured, directly using this data will result in failure of our model to understand some words and it will decrease model accuracy. Thus, we performed the following steps to each tweet we collected.

1. Converted all tweets to lowercase as some text mining algorithms are case sensitive.
2. Browsed through all the tweets and replaced words like - no, not, never, cannot, don't, doesn't with not.
3. Removed repeated words and slang from the sentence and cleaned it.

For example, the sentence is Hiiii! Good to see u then we will remove Hiii and replace it with a word Hi and we will replace word u with you. So, after processing the sentence we will get Hi! Good to see you. We did this using python's nltk tokenizing. After data processing our two example tweets become the following:

Processed Example 1: i wish you would not debate trump unless he debates 1 trump shows taxes debate 2 tells putin stop killing our soldiers debate 3 put sanctions on putin for 2016 election hacks i wish you would publicly challenge him with this biden/rice 2020.

Processed Example 2: best realdonaldtrump political ad I have seen yet. thedemocrats will attempt just about anything to impact the 2020 election. pure evil. cc scavino45 parscale

As we have converted tweets to the trainable tweets, now we can proceed with the model training.

4.4. Model Training

Model training in machine language is the process of feeding an ML algorithm with data to help identify and learn good values for all attributes involved [13]. There are several types of machine learning models, of which the most common ones are supervised and unsupervised learning [14]. We will train the model using Long Short-Term Memory (LSTM) algorithm. LSTM is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feed forward neural networks, LSTM has feedback connections. It can not only process single data points but also entire sequences of data. As we have used Bidirectional LSTM, $\vec{a}^{\langle t \rangle}$ will represent forward and $a^{\leftarrow <t>}$ will represent backward activation.

$$\hat{y}^{<t>} = g(W_y[\vec{a}^{<t>}, \overleftarrow{a}^{<t>}] + b_y)$$

Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data [15].

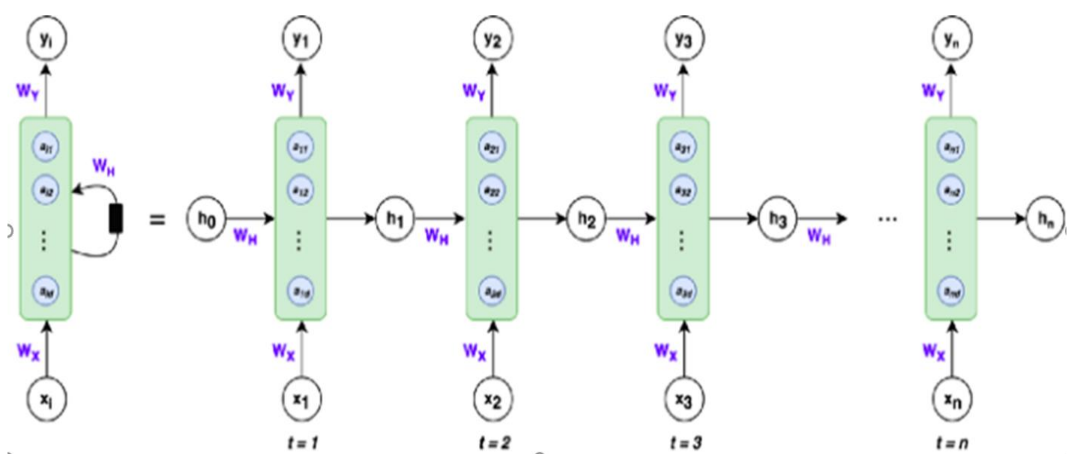

FIGURE 2: Recurrent Neural Network [16].

In a traditional neural network, all inputs and outputs are independent of each other but while doing sentiment analysis input word sequence also matters. In Figure 2 we show the architecture of RNN. Hence, we need to remember the previous work while guessing the next word. So, RNN remembers every information upon time [17]. RNN converts individual activation into dependent activation by providing the same weights to each layer which decreases the increasing parameters and memorizing each previous output by giving each output to the next hidden layer. So now, these hidden layers can be joined together so that the weights and bias of all the hidden layers is the same as a single recurrent layer.

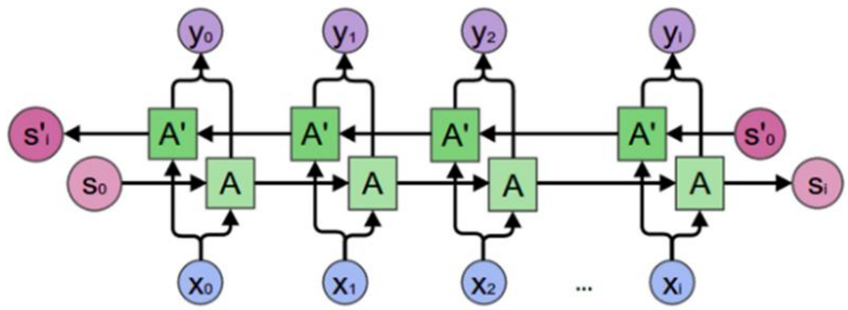

FIGURE 3: Long Short-Term Memory [18].

As shown in Figure 3. Bidirectional recurrent networks are just the same as putting two independent RNNs together. This Bi-LSTM will have two networks, forward and backward information sequences at every step. This will run input in two ways, one is from future to past and another is from past to future. The only difference from bidirectional to unidirectional is that it also runs backward, and you can preserve information from the future using two hidden states. There are some activation functions available like relu, tanh, softmax. We have used relu activation function.

Tuning Parameters	Bi-LSTM Model
Learning Rate	0.01
Dropout	0.5
Embed Size	64
Vocab Size	1000
No Filters	256
Max Length	280

TABLE 1: Model parameters.

We passed English tweets labeled positive or negative to the model. The parameters used in the model are presented in Table 1. Once the training is completed, we saved the model using model.save() which will help us to label the data more rapidly and use the already saved data and there will be no need of saving the data again.

4.5. Testing

Model testing is the process of checking the correctness of the model on the pre-labeled data. To test the model, we passed the pre-labeled data to the saved model. Once the test data is passed to the BiLSTM, the number of parameters is calculated. The number of parameters is the connection between layers and the biases in every layer. The vocab size and max_length is fixed which are 1000 and 280 respectively.

After 25 epochs, the model accuracy is 93.45 % which means that out of 100 sentences, the model will correctly label 93 sentences as positive or negative sentiments. BiLSTM keeps records of previous words and also the next words. For example,

Sentence 1: The person carrying black suitcase is Mr. Mosbey.

Sentence 2: They were shocked when they saw the person carrying water in the desert!

In the above sentences, we cannot say what will be the next word after carrying is black suite case or water. It depends on the context of a sentence. As word orderings are preserved, model accuracy will increase.

Name	Positive %	Negative %
Trump	52.04	47.96
Biden	59.54	40.46

TABLE 2: Sentiment analysis result for 2020 presidential election related tweets.

5. ANALYSIS

For 2020 US Presidential Election predictions, we passed tweets of Trump and Biden that we have collected from October 2019. As shown in Table 2, we got 52% positive sentiment for Trump and 59.5% positive sentiment for Biden. For simplicity, we removed neutral tweets from the consideration and kept only positive or negative sentiment tweets. To have a better understanding of the sentiment, we collected tweets before and after the Covid-19 pandemic for 2020.

We used the same keywords and twitter handles to collect tweets for before and after Covid-19. We took tweets from October 2019 to March 2020 as before Covid-19 tweets and March 2020 to July 2020 as after Covid-19 tweets. The reason behind doing before and after Covid-19 is to analyze how much sentiment has changed in people towards Trump and Biden. Many strategists believe that Trump Administration has failed to stop the Covid-19 in the United States. This means that people are not happy with the Trump administration which can be analyzed by the data we collected.

Name	Positive %	Negative %
Trump (Before)	48.81	51.18
Biden (Before)	54.54	45.36
Trump (After)	49.10	50.90
Biden (After)	66.67	33.33

TABLE 3: Sentiment analysis result before and after Covid-19.

As shown in Table 3, there is a slight increase in people's positive sentiment for Trump. Whereas for Biden, people's positive sentiment for him has increased by 12%. After collecting this data, we collected 2016's data and tried finding some correlation between both the elections.

Name	Positive %	Negative %
Trump	55.73	44.27
Clinton	78.26	21.74

TABLE 4: Sentiment analysis result for 2016 presidential election related tweets.

We have collected 2016's tweet data. The results we got is very appalling. After a significant amount of training and testing, as shown in Table 4 we got 55.73 % positivity for Trump and 78.26 % positivity for Clinton. Usually, positivity denotes that people are in favor of making Clinton the President. But she could not make it and instead Trump won the election to become the President.

To understand this in more detail, we took reference of Donna Ladkin's work [3] where it states the reason why Clinton lost the election even after winning the popular vote by 2.8 million votes in

2016. Trump won the election because Electoral college votes in the States in which he won the popular votes exceeded that of those held by the States in which Clinton won the popular votes. This is the fifth time when a candidate has won by the popular votes but later lost because the Electoral college didn't vote for them.

After comparing both the election data, people have more positive sentiment towards Biden compared to Trump in 2020. So, the possibility of winning the election is higher for Biden as opposed to Trump provided that the Electoral college votes to go for Biden.

6. CONCLUSION AND FUTURE WORK

We have compared the previous work on Machine Learning based Lexicon sentiment analysis with our work on positive/negative words-based Machine Learning algorithm. Though positive/negative words-based Machine Learning algorithm should work well, it failed to give correct sentiment in many cases. So, instead of labelling tweets with positive and negative words, we used libraries to label the tweets and trained model using LSTM. This approach gave better accuracy and reduced the labelling time of each tweet. After that we wanted to find the reason as to why Clinton lost the election in 2016 to Trump despite having much positive sentiment compared to Trump. We discussed and concluded that though public sentiment was much positive for Clinton, the electoral votes didn't go for Clinton in 2016 and thus, she lost the election. In 2020 elections, public sentiment is more positive for Biden compared to Trump. So, if the electoral votes also go for Biden, then he could win the 2020 US General Election.

According to the article [19], there are more republican twitter users to democrats. So, there are chances of getting biased result. So, in order to get more accurate result, we should consider other social media sources like Reddit, Facebook, Blogposts, chat rooms and email campaigns. For these social medias too, our model can be used to predict sentiment for each candidate. To use our model, the data should be in text form (Images, URLs should be removed). And before passing data to the model, the data should be cleaned, structured and labelled.

For future work, we can use official Twitter API to get the more data. GetOldTweets API will fetch very less tweets compare to Twitter Premium APIs. Also, one can use retweets, likes, and share count to weight on other parameters. Also, we can consider keeping track of trending tweets and the social media account getting more tweets in order to consider the current popularity of a candidate. These parameters can help us understand and predict sentiment in better way. For future elections result predictions, our model can be used for sentiment analysis as we have labelled data using VADER API and our model accuracy is 93.45%.

7. REFERENCES

[1] Hao, M., Rohrdantz, C., Janetzko, H., Dayal, U., Keim, D. A., Haug, L.-E., & Hsu, M.-C. (2011). Visual sentiment analysis on twitter data streams. 2011 IEEE Conference on Visual Analytics Science and Technology (VAST). doi:10.1109/vast.2011.6102472.

[2] Tang Y. and Liu J., Gated Recurrent Units for Airline Sentiment Analysis of Twitter Data, Technical Report, Stanford University, 2011.

[3] Ladkin, D. (2017). How did that happen? Making sense of the 2016 US presidential election result through the lens of the ‘leadership moment.’ Leadership, 13(4), 393–412.

[4] Twitter Usage Statistics, https://www.internetlivestats.com/twitter-statistics/, accessed: 2020-08-08.

[5] Social Media on the Campaign Trail: Barack Obama and Donald Trump, accessed on August 8 2020, https://contentgroup.com.au/2017/09/social-media-campaign-trail-obama-trump/.

[6] A. Tumasjan, T. Sprenger, P. Sandner, and I. Welpe. Predicting elections with Twitter: What 140 characters reveal about political sentiment, Proceedings of the International AAAI Conference on Web and Social Media (ICWSM), pp. 178–185, 2010.

[7] A novel classification approach based on Naïve Bayes for Twitter sentiment analysis. (2017). KSII Transactions on Internet and Information Systems, 11 (6). https://doi.org/10.3837/tiis.2017.06.011.

[8] Gautam, G., & Yadav, D. (2014). Sentiment analysis of twitter data using machine learning approaches and semantic analysis. 2014 Seventh International Conference on Contemporary Computing (IC3). doi:10.1109/ic3.2014.6897213.

[9] P. Sharma and T. Moh, "Prediction of Indian election using sentiment analysis on Hindi Twitter," 2016 IEEE International Conference on Big Data (IEEE Big Data 2016), 2016, pp. 1966-1971, doi: 10.1109/BigData.2016.7840818.

[10] B. Joyce and J. Deng, "Sentiment analysis of tweets for the 2016 US presidential election," 2017 IEEE MIT Undergraduate Research Technology Conference (URTC), Cambridge, MA, 2017, pp. 1-4, doi: 10.1109/URTC.2017.8284176.

[11] Vader Sentiment API, https://pypi.org/project/vaderSentiment/.

[12] Jefferson Henrique, Get Old Tweet in Python, https://github.com/Jefferson-Henrique/GetOldTweets-python, accessed: 2020-08-08.

[13] What Is Model Training, https://oden.io/glossary/model-training/, accessed: 2020-08-08.

[14] Alloghani M., Al-Jumeily D., Mustafina J., Hussain A., Aljaaf A.J. (2020) A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science. In: Berry M., Mohamed A., Yap B. (eds) Supervised and Unsupervised Learning for Data Science. Unsupervised and Semi-Supervised Learning. Springer, Cham. https://doi.org/10.1007/978-3-030-22475-2_1.

[15] Long Short-Term Memory, https://en.wikipedia.org/wiki/Long_short-term_memory/.

[16] Khuong, Ben. RNN Figure. The Basics of Recurrent Neural Networks (RNNs). 11 June 2020, medium.com/towards-artificial-intelligence/whirlwind-tour-of-rnns-a11effb7808f.

[17] Khuong, Ben. RNN Definition. The Basics of Recurrent Neural Networks (RNNs). 11 June 2020, medium.com/towards-artificial-intelligence/whirlwind-tour-of-rnns-a11effb7808f.

[18] Aggarwal, Raghav, Bi-LSTM, published in 4 July 2019, medium.com/@raghavaggarwal0089/bi-lstm-bc3d68da8bd0.

[19] How Democrats and Republicans Use Twitter, Pew Research Center, published in Oct 15, 2020, https://www.pewresearch.org/politics/2020/10/15/differences-in-how-democrats-and-republicans-behave-on-twitter/ accessed: 2021-04-15.