Sentiment Analysis of Twitter Data to Predict the Outcome of Mid-Term Elections 2018

Harsha Paruchuri
Department of Computer Science and Engineering
University of California, Riverside
hparu001@ucr.edu

Kaustubh Waghavkar
Department of Computer Science and Engineering
University of California, Riverside
kwagh001@ucr.edu

Zoama Hassan
Department of Computer Science and Engineering
University of California, Riverside
zhass001@ucr.edu

ABSTRACT

Twitter is a widely platform by people for expression of their opinions, on different occasions and in different ways with different intensities. Sentiment analysis is an approach to analyze data and retrieve sentiments that it embodies. Twitter sentiment analysis aims at analysis of sentiments on data from Twitter (Tweets), in order to extract sentiments conveyed by the user. Organizations need to identify the polarity of these opinions in order to understand the user's orientation and thereby make smarter decisions. One such application is in the field of politics, where political entities need to understand public opinion and thus determine their strategies. Sentiment analysis on social media is seen as an effective way to monitor people's inclination towards political parties. In this paper, along with cleaning the data, we have applied sentiment analysis and supervised machine learning principles to the Tweets extracted from Twitter to predict the sentiment of users in their Tweets. We use Logistic Regression model to predict the polarity of the tweets and compare the feature extraction methods of Bag of Words and TF-IDF using our model. After the prediction we calculate the overall sentiment i.e. percentage of positive sentiments and the percentage of negative sentiments in the dataset to forecast the results of 2018 US midterm elections. The final results show that the Democrats have more favorable chances to win a majority in the House.

1. INTRODUCTION

Elections play an important role in politics as they are the means to people's choice of representation and predicting an election outcome has always been interesting. The traditional polls are costly and may fail to make an accurate prediction. In the modern era of the Internet where people express their views through blogs and comments over social media, there is an ocean of opinions, likes, and sentiments of people available and floating around. Analyzing this plethora of free and varied web data of people's sentiments can prove to be a more accurate alternative way to predict election outcomes. This is an interesting research area that combines the two big concerns of today's society- politics and social media. One of the most popular online social networking platforms nowadays is Twitter. Tweets by users are strong indicators of what people talk about or how they feel. Hence, there arises a need to introduce a system that differentiates between good and bad reviews by analyzing the sentiments of tweets and classifying them as positive or negative, to predict the chances of victory of a political candidate or a party. Further, labeling this information with their sentiment would give us an accurate idea about the trend that people follow considering an entity. To cater to this need, we propose to develop an application that would predict the outcome of the upcoming midterm elections to be held in November 2018, by analyzing the sentiments of people exhibited by their tweets.

2. RELATED WORKS

Being an interesting research area, there have been various types of approaches worked upon by researchers. They have found out other ways for predicting election results like relying on volume as stronger indicator than sentiments and using dictionary-based unsupervised methods, multi-stage classification, multi-lingual sentiment classification, human-based tweet classification, decision-tree classifier.

The use of the social networking platform Twitter as a tool to predict the outcomes of social phenomena is a trending practice in recent times. A relevant context which received a lot of attention is the prediction of elections and opinions on political events and decisions. The authors are concerned with predicting electoral and their method aims at exploiting text information in tweets, and most approaches are based on sentiment analysis. They analyze political sentiment in tweets by means of supervised classification with unigram features and an annotated dataset different to and larger than the one we present, achieving 65% accuracy on the task of positive/negative/neutral classification. In this context, sentiment indicates the degree of agreement expressed in a tweet in relation to a political party or candidate. They conclude that volume is a stronger indicator of election outcome than sentiment, but that sentiment still has a role to play. A few studies applied a machine learning approach to classify tweets according to their polarity, either by

training on a manually annotated sample or through dictionary-based unsupervised methods Sentiment analysis methods have been used to improve the predictive results of counting methods, but they still are an open research challenge due for instance to the not trivial identification of sarcasm and irony. [5]
Sentiment analysis identifies and classifies opinions or sentiments which are present in the source text. Social media is generating a huge amount of sentiment rich data in the form of tweets, status updates, reviews and blog posts etc. and analysis of this sentiment data of users is very useful in knowing the opinion of the public. Twitter sentiment analysis is arduous as compared to basic sentiment analysis due to the presence of slang words and misspellings. Machine learning approach can be used for analyzing the sentiments from the text body. Sometimes sentiment analysis is performed by analyzing the twitter posts about electronic products like cell phones, computers etc. It is possible to identify the effect of domain information in sentiment classification using Machine Learning approach. By performing sentiment analysis in a specific domain. They presented a new feature vector for classifying the tweets as positive, negative or neutral and extract people's opinion about products [6].

3. PROPOSED METHOD

We propose a system that aims at predicting the outcome of the Midterm elections held on November 6th, 2018, by gathering the sentiments of people from their posts/tweets on Twitter that are related to this mid-term election and perform a lexicon-based sentiment analysis to predict results. To do this, we design a model that uses Twitter Search API to gather the data for over a period of 7 days before Mid-Term elections 2018. The model cleans the data, finds polarities of all Tweets. Later, the model is trained using a training data set and then predicts the election results. Our hypothesis is that the Democrats have more favorable chances to win a majority in the House.

The expansion of social media in the ongoing past has given end clients a sound platform to voice their opinions. Organizations need to recognize the extremity of these sentiments with the end goal to comprehend client introduction and accordingly settle on more quick-witted decisions. One such application is in the field of legislative issues, where political substances need to comprehend general sentiment and, in this manner, decide their battling procedure.
In [3], the authors propose a two-stage framework which can be used to create a training data set from the mined Twitter information without trading off the content relevance. In the method, the authors perform multistage classification and then identify the polarity of the tweet of the sentiment. The first classifier - an entity classifier used to classify a general stream of data into the respective entities. Next step performs classification based on the polarity of the sentiment w.r.t that of a candidate. Thus, each candidate has a classifier. The whole labelled dataset is used to train the entity classifier; thus, the sentiment is identified pertaining to a candidate. In a nutshell, the authors first tackle the scarcity of
training data for text classification by providing a two-stage framework. Finally, the model for election prediction is proposed, which uses the labeled data created using the framework.
Sentiment Analysis is an assessment of the feeling of the speaker, author or other subject with respect to some point. In US presidential race 2016, Donald Trump, Hillary Clinton and Bernie Sanders were among the best race competitors. The opinion of the general population for a candidate will affect the potential pioneer of the nation. Twitter is utilized to gain a huge diverse data representing the general opinion of the public. The gathered tweets are analyzed using machine-learning based analysis to determine the sentiments of public. In [4], the authors decide the polarity and subjectivity measures for the gathered tweets that assistance in understanding the public opinion for a specific candidate. Further, an examination is made among the candidates over the kind of opinion. Additionally, a word cloud is plotted speaking to most as often as possible showing up words in the tweets. The tools used are Natural Language Toolkit, TextBlob and Twython. In this paper, the authors propose lexicon-based sentiment analyzer which orders the tweets dependent on its sentiment value. The tweets considered are from US presidential decisions 2016. The sentiment classification is done dependent on polarity and subjectivity measures.
These measures imply the positive, negative or neutral attitude towards a specific election competitor, in this manner empowering authors to display the comparison between the best contender for presidential races 2016. Later, a multi lingual based sentiment classification can be worked to obtain tweets in various languages.

4. PROCEDURAL STEPS

This is a software project and our approach involve the following steps:

Data collection using Twitter Search API: The data used is collected through Twitter API by querying a list of keywords related to the elections and the candidates like using keywords like 'Democrats', 'Republicans', 'BlueWave', 'GoRed', 'Midterm', 'Trump', 'Obama', etc. The selection of keywords was large enough to guarantee a good coverage of the elections.
Tweepy python library was used to access the Twitter API data and collected data was dumped into Mongo DB client by establishing a connection between API and Mongo DB

Client. We used Studio 3T to interact graphically with Mongo DB. We fetched around 200k tweets at present. The collected data is in the JSON format and it looks like this:

"_id" : ObjectId("5be26ac2812f573a8c178e2c"),
"created_at" : "Tue Nov 06 23:33:41 +0000 2018",
"id" : NumberLong(1059951988769779712),
"id_str" : "1059951988769779712",
"text" : "RT @IssuesOfMyTime: #Delaware! Vote for progress. Vote for #RobertArlett! An hour and a half left!\n#VoteRed #Midterms2018 #ElectionDay ",
"truncated" : false, "entities" : {
"hashtags" : [
{ "text" : "Delaware",
"indices" : [
NumberInt(20),
NumberInt(29)
}]

Data cleaning to reduce noise in the "dirty" datasets: The preprocessing is a very important step in the project as essential step as it prepares the raw text for mining, making it easier to extract information from the text and apply machine learning algorithms to it. The following steps are executed to clean the dataset:

(a) Removing Twitter Handles: On Twitter, users are acknowledged using Twitter handles, which are of no use and need to be removed. We use a user-defined function to remove the unwanted pattern from the Tweets, which takes two arguments: the original string and the pattern of the text which needs to be removed from the string.

(b) Removing Punctuations, Numbers, and Special Characters: They are removed from the text in the same way as Twitter handles. Everything except characters is replaced with white spaces

(c) Removing short words: Short words with length less than 3 have been removed

Figure (i) shows the tweets before and after cleaning, where the left side shows the raw Tweets and the right side shows the clean Tweets.

RT @tammybaldwin: Students at UW-Madison are v...	student madison vote earli becaus there need w...
RT @RosenforNevada: .@DonaldJTrumpJr, I've liv...	live nevada nearli year been here minut onli a...
RT @Tony4WI: The weekend we've been waiting fo...	weekend been wait final here time vote come do...
RT @RosenforNevada: .@DonaldJTrumpJr, I've liv...	live nevada nearli year been here minut onli a...
RT @tomforwi: I was inspired today by @elizabe...	inspir today honor have warren endors excit
@AmericanProtago @i3onk @GottaSaveBucky @suzan...	http pkiq
RT @Tony4WI: The weekend we've been waiting fo...	weekend been wait final here time vote come do...

Figure i: *Tweets before and after cleaning*

(d) Tokenization: Next, tokenization of all the cleaned Tweets is done in the dataset. This process splits the string of text into tokens

(e) Stemming: The suffixes of a word is being stripped using the rule-based process of stemming. And then the tokens are stitched back together.

(f) Visualization of cleaned Tweets: To gain insight into our data and people's sentiments towards Democrats and Republicans, we have performed certain visualizations using histograms, curves, sentiments.

Extracting features from cleaned Tweets: The dataset is converted into features for further analysis. The assorted techniques of TF-IDF and Bag-of-Words have been used. In Bag-of-Words, text is represented into numerical methods and the build matrix is used as features to build a classification model. The TF-IDF has also been used to penalize the common words by assigning lower weights to them while giving higher weights to words are rare in the collection of texts but appears with high frequency in few documents.

Use of Logistic Regression to predict the polarity of the tweets: We build a model using the training data set that contains clean Tweets with polarities either 'positive' or 'negative' that are manually assigned. We are using Logistic Regression to build the model. It predicts the probability of occurrence of an event by fitting data to a function. The model is trained on the Bag-of-Words and TF-IDF features and it gives the F1 scores.

$$\log\left(\frac{p}{1-p}\right) = \beta_o + \beta(\text{Age})$$

Figure ii: *Equation for Logistic Regression*

We have also used Hyper-Parameter Tuning to get the best parameters to train the model which has significantly improved the performance of the model.

Figure iii: *Code snippet showing the Hyper-Parameter Tuning, Accuracy and Cross-validation*

Testing of Model using polarized test dataset of each political party: We are testing the model with dataset that contains Tweets which are cleaned using the previously mentioned cleaning process.

Predicting the sentiments of Democrats and Republicans: Using the logistic regression model that is built, we predicted the sentiments of Tweets of Democrats and Republicans separately.

Finding the percentage of positive and negative Tweets in the dataset of each political party: After the predictions, we have calculated the percentage of positive and negative sentiments of the Tweets for the datasets.

Comparing percentages to conclude the outcome: We now compare the result percentage to find out which party has more positive sentiments

5. EXPERIMENTAL EVALUATION

5.1 DATA

After tokenization, we get the positive and negative bag of words for Democrats and Republicans which has been depicted in the below histograms. Figure (iv) shows the positive bag of words for the Democrats, Figure (v) shows the negative bag of words for the Democrats, Figure (vi) shows the positive bag of words for Republicans, Figure (vii) contains the negative bag of words for Republicans

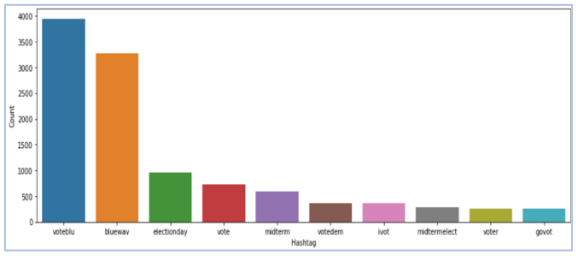

***Figure iv**: Positive bag of words for Democrats*

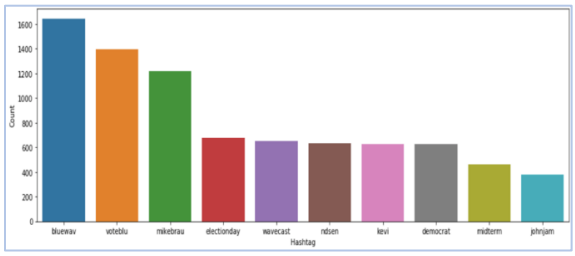

***Figure v:** Negative bag of words for Democrats*

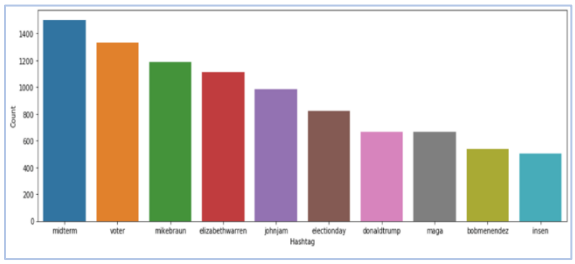

***Figure vi:** Positive bag of words for Republicans*

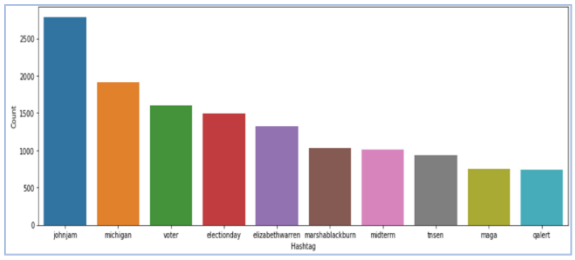

***Figure vii:** Negative bag of words for Republicans*

We have also visualized the Sentiment Maps for the Democrats and Republics datasets. Figure (viii) shows the Sentiment Map for Democrats and Figure (ix) shows the Sentiment Map for Republicans. The largest words are the words that occur with the highest frequency in the dataset

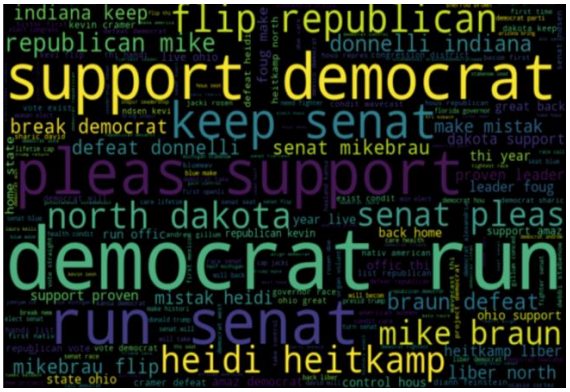

***Figure viii:** Sentiment map for Democrats*

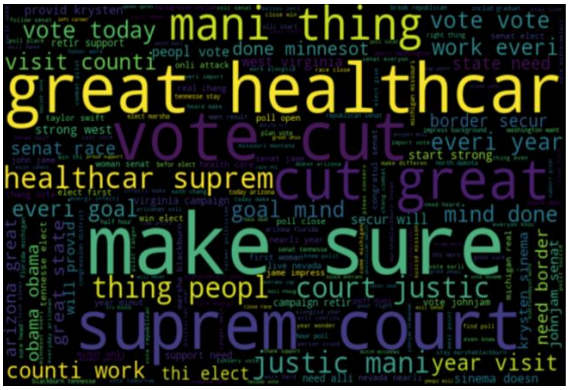

***Figure ix:** Sentiment map for Republicans*

The snapshot of results that show positive and negative polarities for Democrats and Republicans respectively can be seen in Figures (x) and (xi):

```
data1['Polarity'].value_counts(1)*100

1    59.120954
0    40.879046
```

Figure x: *Percentage of Polarity for Democrats*

```
data['Polarity'].value_counts(1)*100

1    63.602763
0    36.397237
```

Figure xi: *Percentage of Polarity for Republicans*

5.2 EXPERIMENTAL SETUP

We have used the following Classification Metrics to evaluate our algorithm:

i) Classification Accuracy:

The classification accuracy for our model is:
87.40% for Democrats and for Republicans is 79.21%.
The code execution for the Classification Accuracy for Republicans can be seen in Figure (xv). The code execution for the Classification Accuracy for Republicans can be seen in Figure (xvi).

ii) Confusion Matrix: This matrix describes the complete performance of our model. The results of this classification metric can be seen in Figure (xiii)

	precision	recall	f1-score	support
0	0.92	0.59	0.72	3280
1	0.85	0.98	0.91	7520
avg / total	0.87	0.86	0.85	10800

Figure xii: *Confusion Matrix*

The ROC curve for true positive rate and the false positive rates are shown in the Figure (xiii) and Figure (xiv)

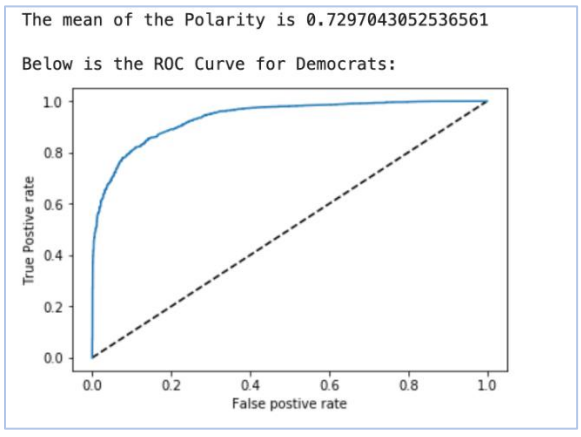

Figure xiii: *ROC Curve for Democrats*

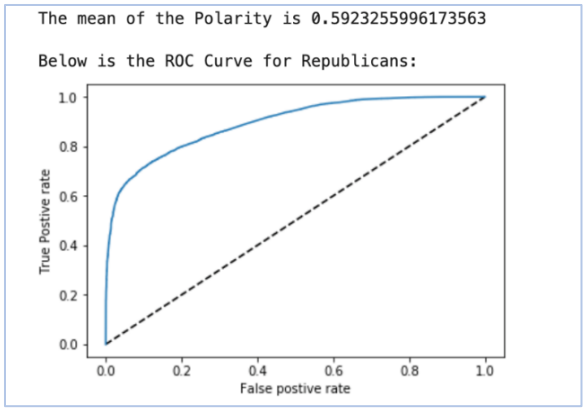

Figure xiv: *ROC Curve for Republicans*

The ROC score for Democrats is found as 0.9357. The ROC score for Republicans is found as 0.8587.

iii) k-fold Cross Validation: We have split our training dataset into 5 subsets and we have trained the model on 4 subsets and tested on 1 subset. The Cross-Validation score for Democrats is 0.8586. The Cross-Validation score for Republicans is 0.76056. The code execution for the Cross-Validation can be seen in Figure (xv). The code execution for the Cross-Validation can be seen in Figure (xvi).

```
accuracy=lreg.score(xvalid_bow, yvalid)*100
print ("The Accuracy of the Algorithm is: ",accuracy,"percent")

The Accuracy of the Algorithm is:  79.20967741935485 percent

#ROC AUC score
from sklearn.metrics import roc_auc_score
print("The ROC Accuracy Score is ",roc_auc_score(yvalid,Y_pred_prob))

The ROC Accuracy Score is  0.893965115684134

#ROC AUC  CV
from sklearn.model_selection import cross_val_score
cv_scores=cross_val_score(lreg,train_bow,train["Polarity"],cv=5,scoring='roc_auc')
print ("The CV Score is: ",cv_scores.mean())

The CV Score is:  0.8586973165907018

#hyperparameter tuning
lreg.get_params()
from sklearn.model_selection import GridSearchCV
from sklearn.cross_validation import train_test_split
c_space=np.logspace(-5,8,15)

#classifier.get_params()
param_grid={'C':c_space,'penalty':['l1','l2']}
logistic_cv=GridSearchCV(lreg,param_grid,cv=5)
logistic_cv.fit(xtrain_bow,ytrain)
print("The Best Parameters Used should be: ",logistic_cv.best_params_)

The Best Parameters Used should be:  {'C': 31.622776601683793, 'penalty': 'l1'}

logistic_cv.best_score_

0.8105529953917051
```

Figure xv: *Code execution for Accuracy of model, ROC Accuracy Score and CV score for Republicans*

```
accuracy=lreg.score(xvalid_bow, yvalid)*100
print ("The Accuracy of the Algorithm is: ",accuracy,"percent")

The Accuracy of the Algorithm is:  87.39814814814815 percent

#ROC AUC score
from sklearn.metrics import roc_auc_score
print("The ROC Accuracy Score is ",roc_auc_score(yvalid,Y_pred_prob))

The ROC Accuracy Score is  0.9357842095874416

#ROC AUC  CV
from sklearn.model_selection import cross_val_score
cv_scores=cross_val_score(lreg,train_bow,train["Polarity"],cv=5,scoring='roc_auc')
print ("The CV Score is: ",cv_scores.mean())

The CV Score is:  0.7605576661419475

#hyperparameter tuning
lreg.get_params()
from sklearn.model_selection import GridSearchCV
from sklearn.cross_validation import train_test_split
c_space=np.logspace(-5,8,15)

#classifier.get_params()
param_grid={'C':c_space,'penalty':['l1','l2']}
logistic_cv=GridSearchCV(lreg,param_grid,cv=5)
logistic_cv.fit(xtrain_bow,ytrain)
print("The Best Parameters Used should be: ",logistic_cv.best_params_)

The Best Parameters Used should be:  {'C': 3.727593720314938, 'penalty': 'l1'}

logistic_cv.best_score_

0.8938095238095238
```

Figure xvi: *Code execution for Accuracy of model, ROC Accuracy Score and CV score for Democrats*

6. DISCUSSION AND CONCLUSION

6.1. Future works

In our future work, we will implement data-debiasing which will involve techniques to ensure that our datasets are free of spam users and bots, etc. which have high potential to influence the working of our model and hence the result prediction. We also plan to work on comprehensive sentiment of Tweets. For this Semantic role labeler can be used which finds out which noun is associated with which verb in the Tweet text.

6.2. Conclusion

By the analysis of public opinions of Twitter, our model has successfully predicted that the Democrats have more favorable chances to win a majority in the House. Our model has achieved an accuracy of 87.40% for Democrats and 79.21% for Republicans. We were fortunate that the time frame of Mid-term elections 2018 overlapped the time-frame of our project implementation. Hence, we were able to compare our prediction with the actual Election results and have been found to be correct based on the actual mid-term election results that came out.

7. ACKNOWLEDGEMENT

The authors thank professor Vagelis Papalexakis for his continuous support and help at each step of the project, guiding us in the right direction and always motivating us to do our best. Work at UCR was supported by University of California, Riverside. We also want to thank the various authors of different research papers whose works were inspirational and helpful for reference.

8. REFERENCES

[1] Jyoti Ramteke, Darshan Godhia , Samarth Shah , Aadil Shaikh.2017.Election Result Prediction Using Twitter sentiment Analysis

[2] Ms. Farha Nausheen, Ms.
Sayyada Hajera Begum. 2018. Sentiment Analysis to Predict Election Results Using Python

[3] Bermingham, A., Smeaton, A.F. 2011. On using twitter to monitor political sentiment and predict
election results

[4] Geetika Gautam, Divakar Yadav.2014. Sentiment Analysis of Twitter Data Using Machine Learning Approaches and Semantic Analysis.

[5] A Razia Sulthana, A K Jaithunbi, L Sai Ramesh.2018.Sentiment analysis in twitter data using data analytic techniques for predictive modelling

[6] Nicolas Tsapatsoulis, Constantinos Djouvas.2017. Feature extraction for tweet classification: Do the humans perform better?

[7] Elvyna Tunggawan, Yustinus Eko Soelistio. 2016. And the Winner is ...: Bayesian Twitter-based Prediction on 2016 U.S. Presidential Election

[8] Suzanne Wetstein.2016. Beating election polls with Twitter A visualization study

[9] Alexander Pak, Patrick Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining", *Proceedings* of the Seventh International Conference on Language *Resources and Evaluation (LREC'10)*, may 2010.

[10] K. Mao, J. Niu, X. Wang, L. Wang, M. Qiu, "Cross-Domain Sentiment Analysis of Product Reviews by Combining Lexicon-Based and Learn-Based Techniques", *2015 IEEE 17th International Conference on High Performance Computing and Communications 2015 IEEE 7th International Symposium on Cyberspace Safety and Security and 2015 IEEE 12th International Conference on Embedded Software and Systems*, 2015.

[11] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, M. Stede, "Lexicon-based methods for sentiment analysis", *Comput. Linguist.*, vol. 37, pp. 267-307, 2011.

[12] C.J. Hutto, E.E. Gilbert, "VADER: A Parsimonious Rule based Model for Sentiment Analysis of Social Media Text", *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*, June 2014.

[13] M. S. Neethu, R. Rajasree, "Sentiment analysis in twitter using machine learning techniques", *Computing Communications and Networking Technologies (ICCCNT) 2013 Fourth International Conference on. IEEE*, 2013.

[14] Mondher Bouazizi, Tomoaki Ohtsuki, "A Pattern-Based Approach for Multi-Class Sentiment Analysis in

Twitter", *Access IEEE*, vol. 5, pp. 20617-20639, 2017, ISSN 2169-3536.
[15] Boia, Marina, "A:) is worth a thousand words: How people attach sentiment to emoticons and words in tweets", *Social computing (socialcom) 2013 international conference*, 2013.
[16] Gao, Wei, Fabrizio Sebastiani, "Tweet sentiment: From classification to quantification", *Advances in Social Networks Analysis and Mining (ASONAM) 2015 IEEE/ACM International Conference*, 2015.
[17] Spencer J, Uchyigit G. Sentimentor: Sentiment analysis of twitter data. InProceedings of European conference on machine learning and principles and practice of knowledge discovery in databases 2012 (pp. 56-66).
[18] Kumar S, Morstatter F, Liu H. Twitter data analytics. New York: Springer; 2014.
[19] Kharde V, Sonawane P. Sentiment analysis of twitter data: A survey of techniques. arXiv preprint arXiv:1601.06971. 2016 Jan 26.