SENTIMENT ANALYSIS OF ENGLISH TWEETS USING BIGRAM COLLOCATION

Sumaya Ishrat Moyeen[1*], Md. Sadiqur Rahman Mabud[2], Zannatun Nayem[2], Md. Al Mamun[3]

[1]*Lecturer, Department of Mechatronics Engineering, Rajshahi University of Engineering & Technology, Rajshahi, Bangladesh*

[2]*M.Sc, Department of Computer Science and Engineering, Rajshahi University of Engineering & Technology, Rajshahi, Bangladesh*

[3]*Professor & Head, Department of Computer Science and Engineering, Rajshahi University of Engineering & Technology, Rajshahi, Bangladesh*

Corresponding Author: Sumaya Ishrat Moyeen

Article DOI: https://doi.org/10.36713/epra8524
DOI No: 10.36713/epra8524

ABSTRACT

Community and portal websites like Twitter, Facebook, Tumbler, Instagram, and LinkedIn etc. have significant impact in our day-to-day life. One of the most popular micro-blogging platforms is twitter that can provide a huge amount of data which in future can be used for various applications of opinion mining like predictions, reviews, elections, marketing etc. The users use this platform to share their views, express sentiments on various events of their daily life. Previously, many researchers have worked with twitter sentiment analysis and compared various classifiers and got the accuracy below 82%. In this work for classifying tweets into sentiments, we have used various classifiers such as Naïve Bayes, Support Vector Machine and Maximum Entropy that segregate the positive and negative tweets. Using Bigram Collocation with classifiers, we've acquired 88.42% accuracy.

KEYWORDS: *Twitter; Sentiment Classification; Machine Learning; NLTK; Python; Naïve Bayes; Support Vector Machine (SVM); Maximum Entropy*

1. INTRODUCTION

In the global and digitalized world, the increasing use of community, social networking and micro-blogging websites and portals, a large volume of data is generated every day. Internet has become an easy-going thing in today's world. It has completely changed the way people thinks and feels of every incidents and issues happening every day. People can share each other’s views and can communicate with each other through online conversation, social media posts and many more. With the advent of deep learning, twitter sentiment analysis has gained higher popularity.

Micro-blogging sites play an important role for gathering huge information. Twitter is one of them which is a social networking platform that gives people access to make tweets for expressing their views and opinions on some topics and issues in their day-to-day life. Every tweet is a live and dynamic content that contains at most 280 characters [1].

In this work, each tweet has a class label in the training data. Different classifiers such as Naïve Bayes, SVM and Maximum Entropy are applied to the training dataset and then the tested tweets are fed into the model. Thus the tweets are classified into positive and negative sets with the help of trained classifiers.

Our aim is to compare performance using various classifiers on twitter dataset. Our proposed method enhances the level of classification by including bigrams. It is clearly seen that "not bad" is a positive expression but the bag of words model could interpret it as negative as it has the word "bad" in it.

For bag of word features, we get better accuracy for SVM when simple train set is used. When we have used bigrams, we got largest accuracy for SVM classifier (88.42%). When comparing between bag of words and bigrams, the accuracy is increased for using bigrams for the classifiers for 10 fold.

The paper is organized as follows: Section 2 states the literature survey, Section 3 describes the framework used which defines various supervised approaches, the used classifiers in the experiments and the concept of using bigram features and cross validation; Section 4 presents our methodology, Section 5 includes results and comparison; and Section 6 draws a conclusion of the work and discusses the future work.

2. LITERATURE SURVEY

Various researches based on sentiment analysis on English tweets have been carried out in the past few years. Some of those are as follow:

- D. Sehgal and A. K. Agarwal (2016) performed Sentiment Analysis on Big Data using HADOOP Framework, Hadoop Database File System (HDFS) and Map reduce and got total accuracy of 72.22% [2].
- S. Kumar, P. Singh and S. Rani (2016) worked with Sentimental Analysis of Social Media Using Rhadoop (R Language and Hadoop) with 200 tweets for R language and 2,000 tweets using Rhadoop [3].
- B. Liu, E. Blasch, Y. Chen, D. Shen and G. Chen (2013) performed Scalable Sentiment Classification for Big Data Analysis Using Naïve Bayes Classifier with Hadoop and got accuracy below 82% [4].
- A.J.Nair, A.Vinayak and V. G focused on logistic regression, VADER and BERT sentiment analysis on COVID-19 tweets where the processing steps will make the algorithms identical [10].
- S.T.Arasteh, M.Monajem, V.Christlein, P.Heinrich, A.Nicolaou, H.N.Boldaji, M.Lotfinia and S.Evert proposed a two-stage deep learning method to create training data with label automatically and the classifier is evaluated on ReTWEET dataset [11].
- G. Saranya, G. Geetha, C.K, M.K and S. Karpagaselvi proposed a system to carry out a continuous sentimental review of the tweets, removed by the twitter [12].

S.B. Kotsiantis highlighted several supervised approaches, perceptron techniques, radial basis function (RBF) networks, heuristic learning algorithms, instance- based learning and Support Vector Machines. Also compared the approaches such as decision trees, single layered perceptrons, multilayered perceptrons, RBF networks, Naïve Bayes classifier and Bayesian networks [5].

3. USED FRAMEWORK

Statistical sentiment classifications based on features are normally experimented on ML algorithms. Mainly the classification falls into two categories, supervised approach and unsupervised approach. The tweeter sentiment analysis problem will be addressed here by supervised learning approach.

3.1 Supervised Approaches

In supervised approach, the data are labeled and both the inputs and outputs are given. The algorithm predicts output from the input and the training is continued until we get good results. At first, the classifiers were trained containing the labeled tweets. Then, the classifiers are used to predict the testing tweets. Hence, the work is done in two steps such as learning and classification. The features were extracted from the tweets and then formed a feature vector in the form of $F = \{f_1, f_2, \dots, f_n\}$ in the learning stage and the labels were combined. The pair of input features indicate the training instance and the expected outcome (sentiment) is denoted by (F, label), where the label is either positive or negative. A new vector is derived known as the training set $TS = \{(F_1, label_1), (F_2, label_2), \dots, (F_n, label_n)\}$. Next, the full training data was used for various ML algorithms for the classifiers to be built efficiently. While classifying, automatically labels are assigned for the testing data set $T = \{F_1, F_2, \dots, F_n\}$ using the trained classifier.

3.2 Used Classifiers

3.2.1 Naïve Bayes Classifier (NBC)

The Naïve Bayesian classifier depends on Bayes' theorem with independent assumptions between indicators. A Naïve Bayesian model is easy to work, with no confused iterative parameter estimation which makes it especially valuable for large datasets. For an example if there are n number of classes C = {C_1, C_2, … , C_n} in a set of documents D = {d_1, d2, … , d_n} and considering W = {w_1,w_2, … , w_s} as unique words set, where each of the word appears at least once in D. Then the probability of d being in class c using Bayes' rule be as follow:

$$P(c|d) = \frac{\mathrm{P(d|c)P(c)}}{\mathrm{P(d)}}$$

Where,

P(c) = prior/ marginal probability of hypothesis c

P(d) = prior/ marginal probability of training data d (Normalization)

P(c|d) = posterior

P(d|c) = likelihood

3.2.2 Support Vector Machine (SVM)

Elements of the training set that changes the position of data points that are closer to the hyper-plane are stated as Support vectors which influence the position and orientation of the hyper-plane. Support Vector Machine is a supervised approach originated from the statistical learning theory. It produces good output even if the the data are complex and noisy. In SVM, the distance between two margins are kept maximum. The surface which is used to separate categories is called hyper plane. The support vectors are considered to be the main elements of the training set [6]. The hyper-plane equation is as follows:

$$g(X) = \omega^T \varphi(X) + b$$

Where,

X= feature vector

w= weight vector

b= bias vector (w and b can be automatically found on the training set)

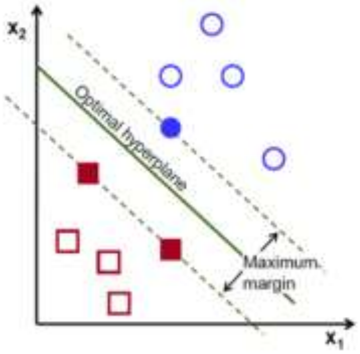

Figure 1. Support Vector Machine [7]

3.2.3 Maximum Entropy Classifier

The maximum entropy classifier is a probability distribution estimation technique. The equation for the Entropy in datasets can be written by,

$$H(p) = -\sum \mathrm{p(label, features)logp(label, features)}$$

If the entropy is maximized, it produces most likely probability distribution:

$$p = argmaxH(p)$$

The probability distribution can be written in the exponential form:

$$p(label|feature) = \frac{1}{z(feature)} \exp(\sum \text{λifi(feature, label)})$$

Where,

f_i (feature, label) = feature function

λi = weight parameter

$z(feature)$ = normalization factor given by:

$$z(feature) = \sum_{label} \exp(\sum_{i} \text{λifi(feature, label)})$$

Here, the Generalized Iterative Scaling (GIS) algorithm is used that scales up well in number of features.

In the theory of Information, the unit Entropy is measured for the unpredictability of the information content. If a dice is thrown randomly, each of the six outcomes have the same probability of occurring (1/6). In this case, the maximum uncertainty occurs with an entropy of 1. If we weight the same dice, the uncertainty becomes low. If we weight the dice to such an extent that the outcome is always six, the uncertainty becomes zero and hence the entropy of information content is also zero.

3.3 Bigrams

When we are working with text classification, unigrams are considered as single words indicating features. Before applying bigrams in the bag of words model, first the unigrams were extracted, then their frequency is counted and if the frequency is more than one, then the word is added as a feature and the value is set as its frequency.

Feature vector is used to construct the set of bigrams. In some situations, a negation can be added to a word like a "no" or a "not", maybe before or after the word. It is a collection of n consecutive words used as a building block. For example, if we consider the sentence "I have not done yet!" bigrams take it as "I have+not" and "have+not done". Negation can improve the accuracy of live sentiment classification. Bigrams generate better classifications by improving the performance. For a long phrase trigrams can be used otherwise it'll lead to lower the performance. That's why we've considered only unigrams and bigrams. Tweets need to be preprocessed before generating the feature vectors from words.

3.4 Cross Validation

Different random state values generate unstable accuracy for machine learning model that's why cross validation is important. The sample dataset is divided into n number of subsets for an n-fold cross validation. The method using the n subsets as the test set or validation set and the other (n-1) subsets as training set can increase the accuracy by avoiding over fitting.

4. PROPOSED METHODOLOGY

4.1 Tools (NLTK)

The Natural Language Toolkit (NLTK) is used for tokenization, stemming and tagging in sentiment analysis process.

4.2 Twitter Application Management

To access the public data from the twitter, it provides twitter API to developers which extracts features from the tweets for sentiment mining. As a new user completing the registration process using OAuth, a token is given by twitter to connect it's database actually Twitter Developer Labs endpoints. Total 1000 tweets are retrieved including 500 positive and other 500 negative tweets using provided credentials by Twitter. The process is done by accessing Twitter API, setting request parameters and filter method using Python code.

4.3 Preprocessing

The data generated from the twitter using streaming API contain various website links, emoticons, white spaces, hash-tags etc. which contain no sentiment at all and that's they should be removed before processing it so that the sentiment generated is accurate enough.

At the start of the preprocessing, all the tweets are converted to lower case. The tweets containing any URL, special symbols such as punctuations, hash-tags, additional white spaces has been removed. Example: "How beautiful the day is!" is replaced by "The day is beautiful".

At present, emoticons play a vital role in expressing the sentiments. They are replaced with the corresponding word to do the analysis efficiently. Next, stemming is done to convert a word to its grammatical root form like "walk", "walker", "walking", "walked", "walks" to root word "walk".

Stop-words are filtered as they are mostly considered useless. Without increasing the precision or recall the amount of the index rises exponentially. NLTK has a list of built-in stop-words that contains 128 English stop-words.

Finally, feature extraction is performed for dimensional reduction which only contain the relevant information of the data by removing the redundancy. In this work, it finally returns a dictionary. We take both positive and negative lists and create a single list. On the list the first element is the array of features and second element is the sentiment label. To have better classifier accuracy, precision and recall the improvement of feature extraction is needed. Two modifications are used here: filtering out the stop-words and including bigram collocations.

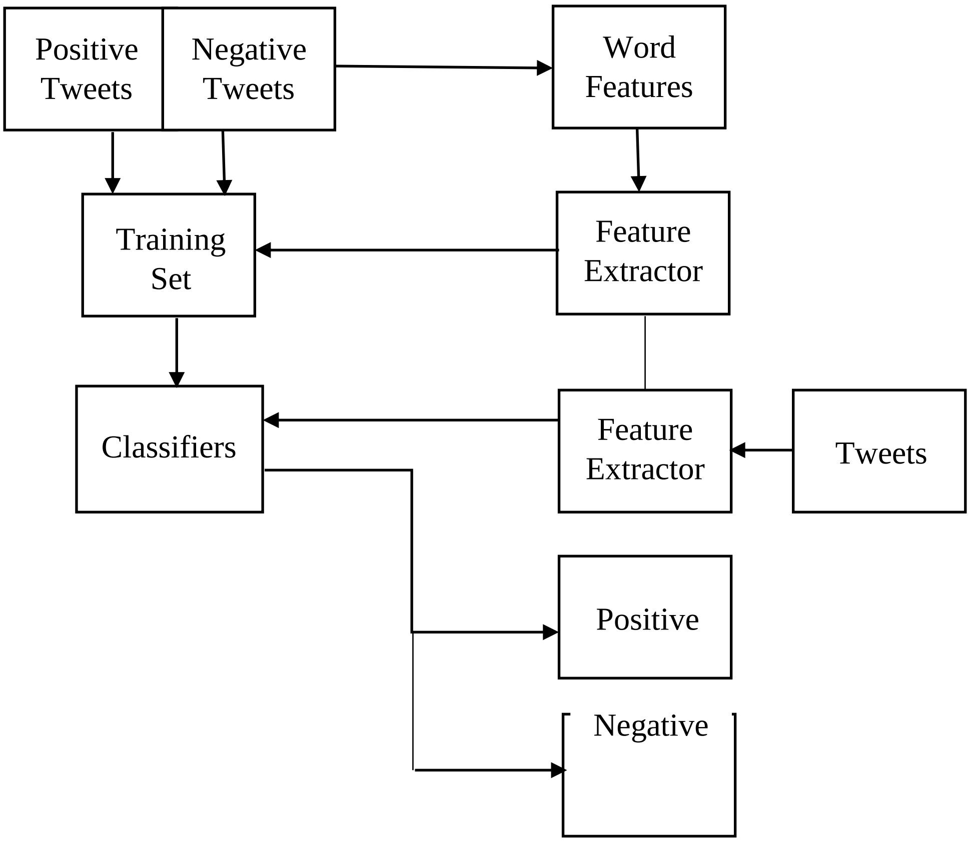

Figure 3. Proposed System Architecture

5. RESULT AND COMPARISON

For bag of word features without stopwords, we get better accuracy for SVM when simple train set is used and the accuracy is 88.4%. When we have increased the fold to 10 in cross validation, the accuracy has increased for Naïve Bayes and Maximum Entropy Classifiers but has decreased in SVM.

With stopwords, again we get better accuracy for simple train set SVM but the accuracy decreases in comparison with the previous one (without stopwords) and the accuracy is 87.6\%. But for Naïve Bayes and Maximum Entropy Classifiers, the accuracy is increased after combining stopwords with bag of words. When we have increased the fold to 10 in cross validation, the accuracy has decreased for Naïve Bayes Classifier in comparison with SVM and Maximum Entropy Classifiers.

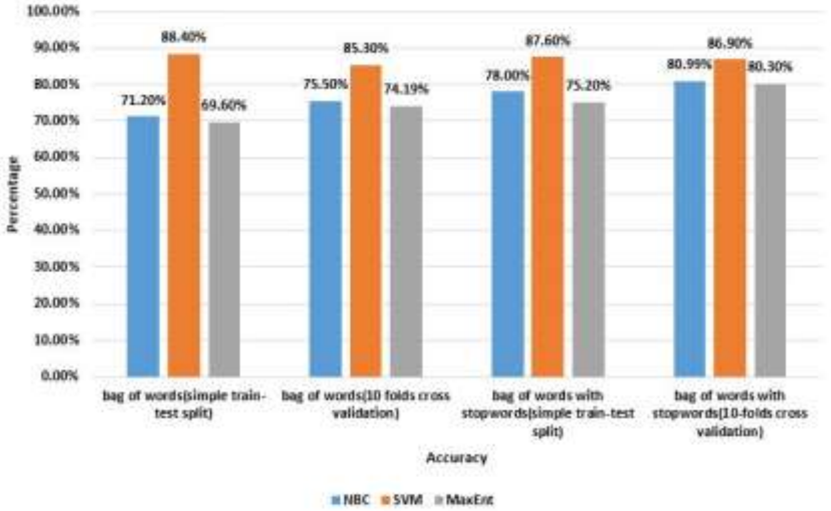

Figure 4. Comparison of Bag of Words with and without stopwords

When we have used bigrams, we got largest accuracy for SVM classifier (88.42%). When comparing between bag of words and bigrams, the accuracy is increased for using bigrams for the classifiers for 10 fold.

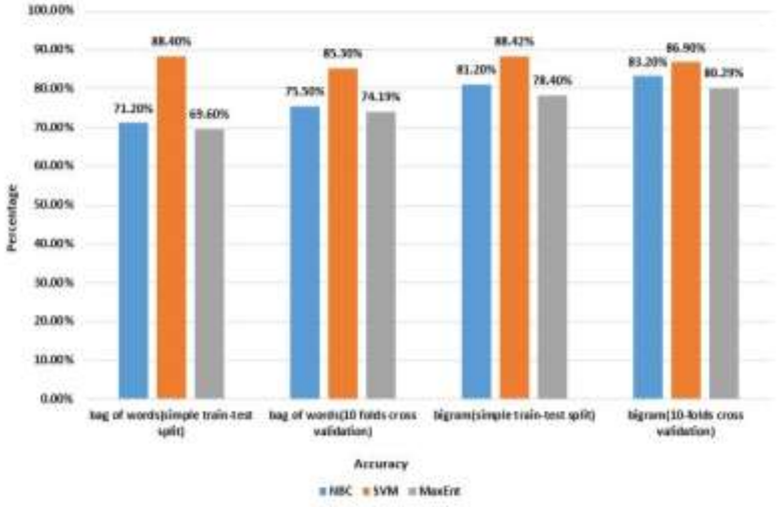

Figure 5. Comparison of Bag of Words and Bigrams without stopwords

When stop word is combined with bigrams, we get even more accuracy. This time, again the SVM has the largest accuracy. But in 10 fold, the Naïve Bayes classifier has the largest accuracy (85.8%).

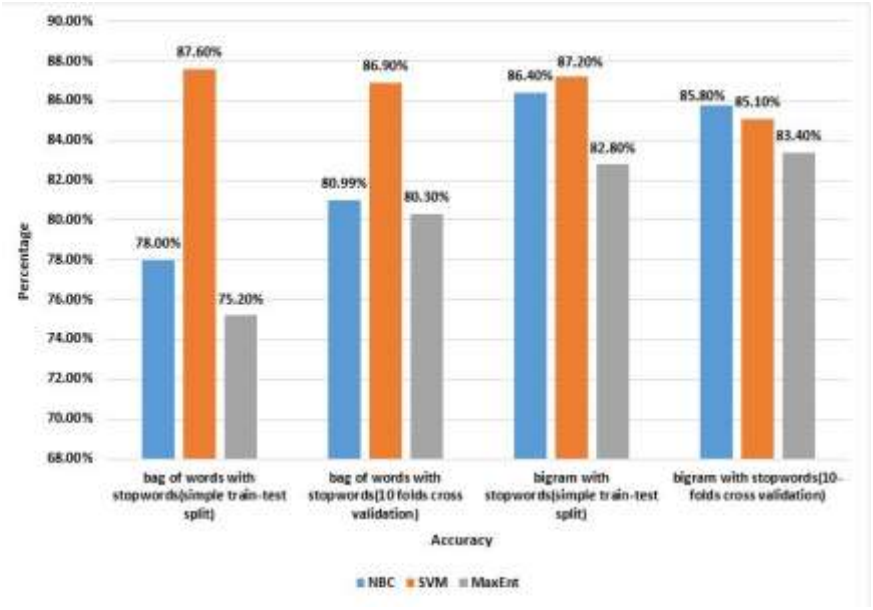

Figure 6. Comparison of Bag of Words with stopwords and Bigram with stopwords

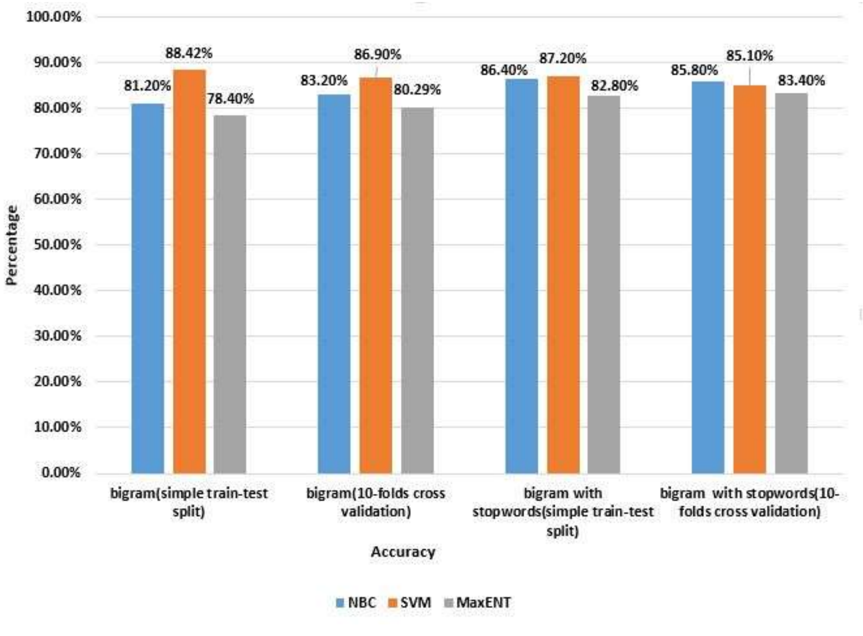

Figure 7. Comparison of Bigrams with and without stopwords

From the above mentioned comparison, we show that SVM gives better accuracy for bigram features and the accuracy is 88.42% but the execution time has increased. Bag of word feature gives better execution time then bigram feature.

6. CONCLUSION AND FUTURE WORK

In this work, we've proposed a method to classify tweets into positive and negative sentiment. We've collected huge amount of data for a specific topic from twitter via Twitter streaming API, extracted features from Tweets and classified tweets using several classifiers. Sometimes some words are combined together in a different way thinking outside the box, which makes it really hard for defining sentiment. This work can be improved by using parser embedded system, dealing with sentences of multiple meanings, increasing the classification categories, working on multiple languages and speeding up the classifiers. Here, only English tweets have been taken into consideration which can be expanded in future.

REFERENCES

1. *Twitter (2015) Tweet updates [Online]. Available: https://developer. twitter.com/en/docs/tweets/tweet-updates. [Last visited on: 10- Oct-2019].*
2. *D. Sehgal and A. K. Agarwal, "Sentiment analysis of big data applications using Twitter Data with the help of HADOOP framework," 2016 International Conference System Modeling & Advancement in Research Trends (SMART), pp. 251-255, Moradabad, 2016.*
3. *S. Kumar, P. Singh and S. Rani, "Sentimental analysis of social media using R language and Hadoop: Rhadoop," 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), pp.207-213, Noida, 2016.*
4. *B. Liu, E. Blasch, Y. Chen, D. Shen and G. Chen, "Scalable sentiment classification for Big Data analysis using Naïve Bayes Classifier," 2013 IEEE International Conference on Big Data, pp. 99-104, Silicon Valley, CA, 2013.*
5. *S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques,"2007 conference of Emerging Artificial Intelligence application in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies, pp. 3-24, 2007.*
6. *Support Vector Machine Background for Feature Extraction (2010) [Online]. Available: http://www.harrisgeospatial.com/docs/ BackgroundSVM.html. [Last visited on: 19-sep-2019].*
7. *L. Mandloi and R. Patel, "Twitter Sentiments Analysis Using Machine Learning Methods," 2020 International Conference for Emerging Technology (INCET), pp. 1-5, Belgaum, India, 2020.*
8. *E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the omg!,"Icwsm, pp. 538:541, vol. 11, 2011.*
9. *Q. King. (2014) Sentiment Analysis of Big Data with Apache Hadoop [Online]. Available: https://scholarspace.manoa.hawaii.edu/bitstream/10125/101227/1/KangQiulingr.pdf. [Last visited on: 10-April-2019].*
10. *A. J. Nair, V. G and A. Vinayak, "Comparative study of Twitter Sentiment On COVID - 19 Tweets,"2021 5th International Conference on Computing Methodologies and Communication (ICCMC), pp. 1773-1778, Erode, India, 2021.*
11. *S.T.Arasteh, M.Monajem, V.Christlein, P.Heinrich, A.Nicolaou, H.N.Boldaji, M.Lotfinia and S.Evert, "How Will Your Tweet Be Received? Predicting the Sentiment Polarity of Tweet Replies," 2021 IEEE 15th International Conference on Semantic Computing (ICSC), pp. 370-373, CA, USA, 2021.*
12. *G. Saranya, G. Geetha, C.K, M.K and S. Karpagaselvi, "Sentiment analysis of healthcare Tweets using SVM Classifier," 2020 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), pp. 1-3, Chennai, India, 2020.*