Sentiment Analysis On Twitter Data Using Machine Learning Algorithms

S.Vijaykrishnaa[1], M.P. Geetha[2], D.S.K.Vinoth[3], S.Pavithra[4]
[1, 2, 3, 4] Dept of Computer Science and Engineering
[1, 2, 3, 4] Sri Ramakrishna Institute of Technology Coimbatore, Tamilnadu, India

Abstract- *Nowadays the living style of people has changed due to the enormous advancement of the Internet. People express their emotions and views through social media. Twitter is a micro-blogging website that allows people to share and express their views about topics, or post messages. Currently Twitter has approximately 145 million daily active users and daily generates nearly 500 million tweets per day. Due to this large amount of usage, we hope to achieve a reflection of public sentiment by analyzing the sentiments expressed in the tweets. Analyzing the public sentiment is important for many applications such as firms trying to find out the response of their products in the market, predicting political elections and predicting socioeconomic phenomena like stock exchange. In this project, we proposed a web app which shows sentiment analysis results and its visualizations. Based on the keyword that the user inputs, the relevant real time tweets were retrieved, and the sentimental analysis is done to produce the results.*

Keywords- Cross Validations, Machine Learning, Sentiment Analysis, Support Vector Machine, Twitter

I. INTRODUCTION

With an exponential rise in social media usage to share emotions, thoughts and opinion, Twitter has become the goldmine to analyze brand performance. Opinions found on Twitter are casual, honest and informative than what can be picked from formal surveys etc. Millions of users express their sentiment about brands they interact with. These sentiments, if identified, can be useful for companies to not only monitor their brand's performance but also locate aspects and time periods that receive polar sentiments. These brands can be products, celebrities, events or political parties.

However, with more than 500M tweets sent per day, this data has already become huge enough to be analyzed by any team manually. Likewise, the diversity of tweets presumably cannot be captured by a fixed set of rules designed by hand. It is worth noting that the task of understanding the sentiment in a tweet is more complex that of any well formatted document.

The features that can be used for modeling patterns and classification can be divided into two main groups: formal language based, and informal blogging based.

Tweets do not follow any formal language structure, nor they contain words from formal language (i.e., out of vocabulary words).

Often, punctuations and symbols are used to express emotions (smileys, emoticons etc.). In this work, we use Machine Learning (ML) approach and Natural Language Processing (NLP) techniques to understand the patterns and characteristics of tweets and predict the sentiment they carry. Performing natural language processing on textual data from Twitter presents new challenges because of the informal nature of this data. Tweets often contain misspellings, and the constrictive limit of 140 characters encourages slang and abbreviations. Unconventional linguistic means are also used, such as capitalization or elongation of words to show emphasis.

Additionally, tweets contain special features like emoticons and hashtags that may have an analytical value. Hashtags are labels used for search and categorization and are included in the text prepended by a "#". Emoticons are expressions of emotion and can either be written as a string of characters e.g., ":-)", or as a Unicode symbol. Finally, if a tweet is a reply or is directed to another Twitter user, mentions can be used by prepending a username with "@". Specifically, we build a computational model that can classify a given tweet as either positive, negative or neutral based on the sentiments it reflects.

A positive and negative class would contain polar tweets expressing a sentiment. However, a neutral class may contain an objective or subjective tweet either a user reflects neutrality in an opinion or contains no opinion at all. The decision to use three classes is made to accommodate the complexity of the problem and is consistent with ongoing research in the field. Using our sentiment predictor, we can also build an interactive visualization tool to help businesses interpret and visualize public sentiments for their product and brands.

This tool enables the user to not only visualize plain sentiment distribution over the entire dataset, but also equips the user to conduct sentiment analysis over the dimension of time, location, and influencing power of a user. Before understanding research conducted for Twitter sentiment analysis, we need to describe standard procedure to tackle this problem.

Sentiment analysis has been traditionally tackled as a classification task (supervised learning) where the user decides which classification algorithm to use. Supervised Text Classification, a machine learning approach where a class-predictor is inferred from labeled training data, is a standard approach for sentiment classification of tweets.

II. LITERATURE SURVEY

Pert et al. [1] proposed a deep RNN classifier based on Firefly based Water wave optimization algorithm for sentiment analysis. Their objective was to develop an optimization algorithm for sentiment classification. They used an open dataset named twitter airline sentiment from Kaggle website for their research. It showed better results as compared to the performance by other deep learning neural networks. Their model can be extended by mapping the Socioeconomic data with the Sentiments from Twitter Sentiment Analysis.

Ruz et al. [2] proposed a Bayesian network classifiers approach which includes Naive bayes (NB), tree augmented Naive Bayes (TAN), Support vector machine (SVM), BF TAN, and RF. Their objective was to address the problem of sentiment analysis during critical events such as natural disasters or social movements. They extracted tweets with hash-tags #terremoto chile, 60,000 tweets from the 2017 Catalan independence referendum for dataset. The resultant model identifies the relations amongst words, offering interesting qualitative information too. The drawback of their model is, it heavily relies on quality of the training data and heavily time and event dependent.

Plunz et al. [3] proposed an Ensemble approach in which Naive Bayes Logistic Regression with embedding features (NBLR + POSwemb model) is used. Their objective was to assess if tweets generated in parks may express a more positive sentiment than tweets generated in other places in New York City. For 549 days between June 17th, 2016 and December 17th, 2017, tweets were collected by using the filter method of Twitter's streaming API. A python wrapper called tweepy was used to handle the connection. The only filter provided in the query was location. They were useful in interrogating the widely held belief that urban parks contribute to general well-being of residents in cities. The drawback of their model is the use of geolocated social media data and small coverage of the area.

Soumya et al. [4] Conducted a study on different Machine learning techniques that can be used for Sentimental analysis. Their objective was to do Sentiment Analysis on Malayalam tweets, which have been classified into positive and negative using different machine learning algorithms. They used Naive bayes (NB), Support Vector Machine (SVM), Random Forest (RF) for their research. They created a dataset by retrieving tweets using twitter API, twenty-two (22) positive and 13 negative Malayalam words are identified and used as the hashtag for retrieving tweets. Their results showed better accuracy in prediction. The drawback of their model is the use of only language features.

Akilandeswari et.al [5] proposed a Scoring Model Technique for twitter sentiment analysis. Their objective was to Design a scoring model incorporating language and non-language features. They extracted English language tweets to create a dataset by using twitter API. The model enhances the accuracy of the assignment of polarity to tweets. Their model can be extended to incorporate suitable statistical techniques to analyze the classification performance.

Chiarello et.al [6] proposed a tandem LSTM-SVM classifier to describe the Advantages or Drawbacks of products as an effect of the interaction between artifacts and users using twitter data. The Twitter Streaming API is used to extract tweets using keywords from 11th June 2017 hr. 10:00 to 31st July 2017 hr. 15:00. The obtained dataset is then filtered by removing tweets with less than five words and non-English posts with a language classifier. After that, an SVM model was used for predicting the probability for each tweet of being relevant or irrelevant with a threshold of 0.7. The final dataset of filtered tweets is made up of 66,796 posts. In their work, they used Keras deep learning framework and LIBSVM to generate respectively the LSTM and the SVMs statistical models. It can be further used to develop quantitative measures of the difference between the collections obtained using the two approaches, using metrics from Information theory and validating them across several experiments.

Ansari et.al [7] attempted a supervised learning problem where they mined tweets to capture the political sentiments from it. Their objective was to analyze the people's opinion and predict the future trends in the elections. They collected 3896 relevant tweets by querying Twitter with keywords like party names, their abbreviations, and by the names of their most popular leaders along with the election

hashtag, #LokSabhaElections, #ElectionsInIndia, etc. The Long Short Term Memory (LSTM) is employed to prepare the classification model and compared it with Support Vector Machines, Decision Tree, Logistic Regression, and Random Forest algorithms. From output, they came to know that promising results are achieved with LSTM and Random Forests algorithms. The model can be further improved by increasing the number of tweets extracted and proper sampling methods must be incorporated in order to balance the class distribution among all the classes.

El_Rahman et.al [8] proposed a lexicon-based classification approach for twitter sentiment analysis. Their objective was to see who people think is better McDonalds or KFC in terms of how good/bad reviews are and to show which restaurant has more popularity. The Dataset had been extracted directly from Twitter API and were used to train and test the models. Finally, for McDonald's 7000 tweets and KFC 7000 tweets were extracted. A lexicon-based classifier used a manually created lexicon to find the sentiment of each tweet. They used multiple supervised learning algorithms for the purpose of training: Naive Bayes, support vector machine (SVM), maximum entropy, decision tree, random forest and bagging. The prediction showed improvements in comparison to existing work where a label data is present.

Sharma et.al [9] attempted both lexicon based and NRC Dictionary Based Approach for Sentiment analysis. Their objective was to gain the opinion polarity of the folks concerning general elections held in India. Twitter's API is being used to collect the tweets of the two respective candidates. They used R, Rapid-Miner AYLIEN for Sentimental analysis. In future some additional data like pictures, sounds, online articles, multimedia etc. could be added to this data set. In this work, geospatial information like longitude, latitude is completely ignored. In future all these characteristics would be considered. Another future recommendation would be to use hashtags on the top of the tweet sentiments as features for classification of the text.

Das et.al [10] proposed a deep learning approach using RNN components from the Stanford Core NLP software. Their objective was to predict the potential prices of a company's stock by analyzing day-to-day streaming tweet's sentiment. Actual Stock prices of Google, Microsoft and Apple have been collected using Yahoo Finance website. The final dataset procured contains 5,60,000 tweets which range over the span of thirteen years of twitter data starting from May 2005 till Jun 2017. The results obtained can be used to eventually forecast movements of individual stock prices. In future, inclusion of eclectic machine learning algorithms can be done for the prediction of stock data such as Deep learning models.

III. PROPOSED METHODS

This section explains about the Dataset, Preprocessing Methods, Feature Selection, and Classifiers used in our experimental setup.

A. Dataset

We used a dataset containing 1,600,000 tweets extracted using the Twitter API in English Language from Kaggle website named "Sentiment140". Here the tweets have been annotated (0= negative, 4 = positive). For my experiment, I have used a NLP package named "Text Blob" which helped me to label the dataset into positive, Negative and Neutral tweets.

B. Preprocessing

Tweets are unstructured text, which make them difficult to score accurately. Although each tweet is only 140 characters, they're filled with links, acronyms, emoticons, misspelled words, slang words, and much, much more. It is very important since all the modifications that we are going to do during this process will directly impact the classifier's performance. The pre-processing includes cleaning, normalization, transformation, feature extraction and selection, etc. The result of pre-processing will be consistent and uniform data that are workable to maximize the classifier's performance. Some of the preprocessing steps done are:

- Removing twitter usernames and real names (@handles)
- Removing punctuation, numbers and Special characters
- Removal of commonly used words (stop words)
- Removing Short words (less than three letters)
- Removing White Spaces
- Removing HTML tags and URL links, dates (if present)
- Replacing acronyms with their actual words
- Replacing the emoticon encoded values with their actual meaning
- Replace other cases to lower case letters
- Replacing Negation words with not

C. *Feature Extraction*

Bag of Word, Term Frequency vs. Inverse Document Frequency has been considered for feature vector formation of the input dataset.

- ***Bag of Words (BOW):*** In BOW, the text is transformed into a bag of words where each entry corresponds to the number of occurrences of a particular term in the sentence. The feature matrix is created with m * n dimension where m is the number of sentences and n is the number of unique words in the corpus [4].

- ***TF−IDF:*** TF−IDF is the statistical measure to evaluate the significance of a particular term in a corpus.

$$tf - idf = tf \square * idf \square$$

where tf is the term frequency and idf is the inverse document frequency [4].

D. *Machine Learning Classifiers*

Three different Machine Learning Classifiers such as LR, NB, SVM were applied for predicting the sentiment of the tweets.

- ***Naive Bayes Classifier:*** NB predicts the sentiment of the test dataset as positive or negative using a Multinomial NB classifier [4]. This classification is done based on Bayes' theorem. It is regarded as one of the most suitable for word counts.

- ***Support Vector Machine (SVM):*** SVM is a supervised machine learning algorithm which is widely regarded as one of the best text classification algorithms was proposed by Vapnik in 1992. It finds the linear separator with maximum marginal distance using support vectors in high dimensional space [4]. Here both linear kernel function and stochastic gradient descent (SGD) learning method have been used for predicting the sentiment of tweets and the results were compared.

- ***Logistic Regression (LR):*** LR is a machine learning Classifier which classifies based on probability prediction. Here the probability prediction must be transformed into a binary value using the logistic function.

IV. IMPLEMENTATION

First, the twitter dataset was labeled positive, negative, or neutral using the Text Blob Python package and the labeled tweets were preprocessed. Then, a feature vector is formed for each labeled tweet where the values of this vector will characterize the sentiment.

Once feature vectors are extracted for each tweet in the labeled dataset, they are fed a classification algorithm that attempts to find relations between each value (called feature) in the vector and the labeled sentiment. The feature sentiment relationship is captured by these algorithms and stored in a learned model. When a new instance is provided to the model, it uses already learned relationships to predict the sentiment.
Here we have used *CountVectorizer* which tokenizes, filters stop words and builds a dictionary of features and transforms processed text data to feature vectors. It supports each fold, the training data is divided into 16 splits (15 training splits and 1validation splits) where each split has 1,00,000 tweets in it.

The "Precision" metrics attempts to answer the proportion of Positive Identification which was correct. It can also be defined as the number of True Positives divided by the number of True Positives and False Positives.

$$Precision = \frac{TP}{TP + FP}$$

counts of N-grams of words or consecutive characters. To normalize this feature vector, we divide the number of occurrences of each word in a tweet dataset by the total number of words in the tweet dataset. These new features are called *'tf'* for *Term Frequencies*.

The main reason to normalize is that longer text will have higher average count values than shorter text, even though they might talk about the same topics. Another refinement on top of *'tf'* is to downscale weights for words that occur many times in tweets corpus and are therefore less informative than those that occur only in a smaller portion of the corpus. This downscaling is called *'tf–idf'* for *Term Frequency times Inverse Document Frequency*.

We have used the Cross Validation method on the dataset to avoid overfitting and to find the best classifier which has higher accuracy and minimum overfitting over data. Here for

The "Recall" metrics attempts to answer the proportion of Actual Positives which was correctly identified.

It can also be defined as the number of True Positives divided by the number of True Positives and the number of False Negatives.

$$Recall = \frac{TP}{TP + FN}$$

The F-score or F-measure (F1-Score) is the measure of a test's accuracy. It is calculated from the precision and recall of the test.

$$F1 = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall}\right)$$

The Support is the number of actual occurrences of the class in the specified dataset.

Table 1
Classification Report for Naive Bayes

Sentiment	Predicted Value	Precision	Recall	F1-Score	Support
Negative	-1	0.95	0.59	0.73	98
Neutral	0	0.92	0.71	0.80	167
Positive	1	0.74	0.97	0.84	232
Accuracy				0.81	497
Macro Avg		0.87	0.76	0.79	497
Weighted Avg		0.84	0.81	0.81	497

Table 2
Classification Report for Logistic Regression

Sentiment	Predicted Value	Precision	Recall	F1-Score	Support
Negative	-1	0.97	0.94	0.95	98
Neutral	0	0.98	0.98	0.98	167
Positive	1	0.98	0.99	0.99	232
Accuracy				0.98	497
Macro Avg		0.98	0.97	0.97	497
Weighted Avg		0.98	0.98	0.98	497

Table 3
Classification Report for SVM (Linear Kernel)

Sentiment	Predicted Value	Precision	Recall	F1-Score	Support
Negative	-1	0.98	0.95	0.96	98
Neutral	0	0.99	0.99	0.99	167
Positive	1	0.97	0.99	0.98	232
Accuracy				0.98	497
Macro Avg		0.98	0.97	0.98	497
Weighted Avg		0.98	0.98	0.98	497

Table 4
Classification Report for SVM (SGD Method)

Sentiment	Predicted Value	Precision	Recall	F1-Score	Support
Negative	-1	1.00	0.45	0.62	98
Neutral	0	0.72	0.96	0.82	167
Positive	1	0.90	0.89	0.89	232
Accuracy				0.82	497
Macro Avg		0.87	0.76	0.78	497
Weighted Avg		0.86	0.82	0.81	497

V. RESULT & DISCUSSION

The Classification Report have been generated for NB, SVM (both Linear and SGD methods), and LR classifiers which contains Precision, Recall and F1-Score measures. Feature matrix with BOW, and TF− IDF are created by considering all the unique words in the corpus. The cross-Validation scores for each model along with respective folds have been generated below.

From the experiment, it is found that the Support Vector Machine with linear kernel had higher accuracy than the other models. The Linear Regression model had accuracy slightly lesser than the SVM Linear kernel.

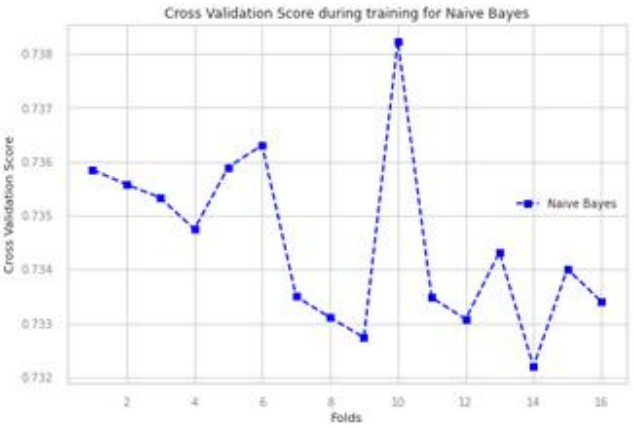

Fig 1. Cross Validation Scores of Naïve Bayes

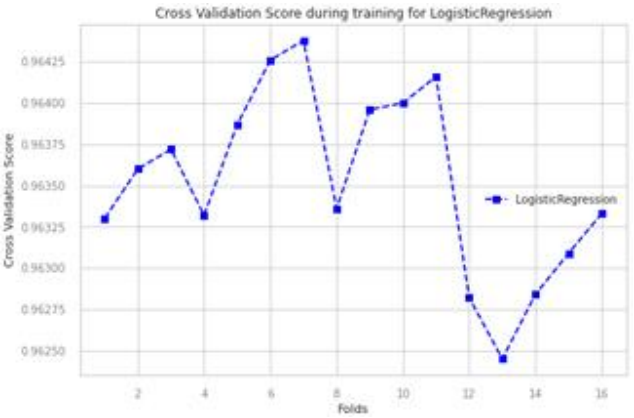

Fig 2. Cross Validation Scores of Logistic Regression

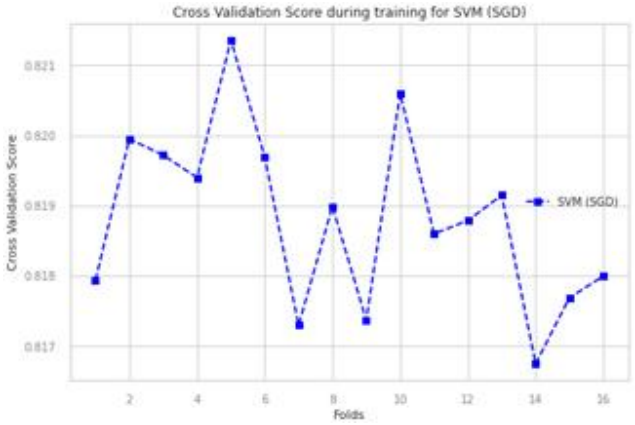

Fig 3. Cross Validation Scores of SVM (SGD)

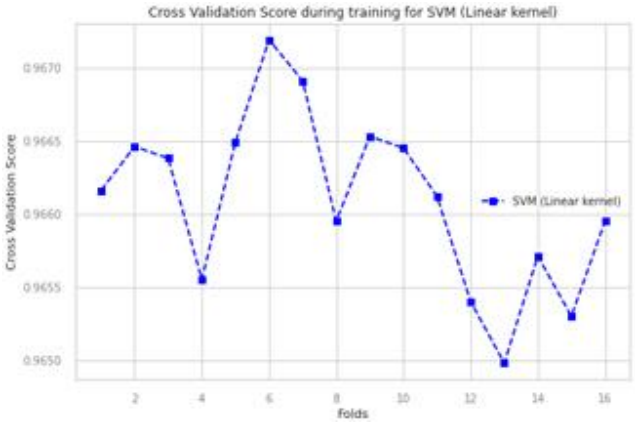

Fig 4. Cross Validation Scores of SVM Linear Kernel

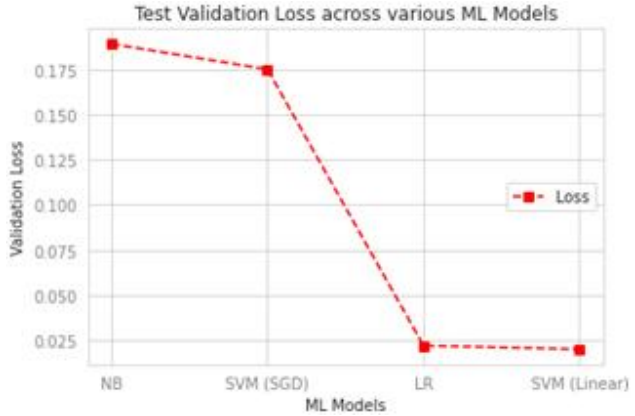

Fig 5. Validation Losses for various ML models on test data

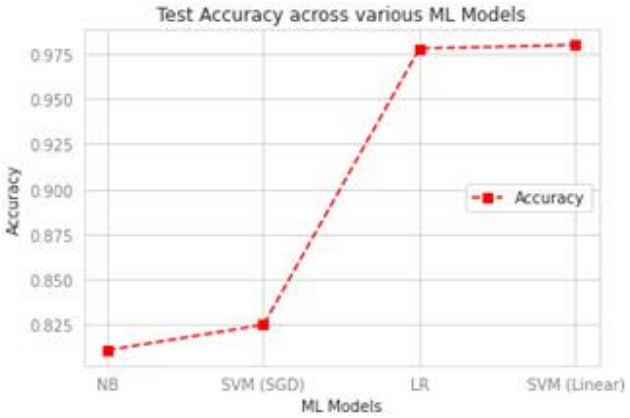

Fig 6. Accuracy of various ML models on test data

VI. CONCLUSION

Sentiment Analysis of English tweets using Naïve Bayes, Linear Regression, Support Vector Machine models were proposed in this work. Bag of Words, Tokenization, Term Frequency – Inverse Document were used for feature extraction purposes. The Python Packages "RE" and "TEXT BLOB" were used for preprocessing of tweets in the dataset. Cross Validation methods have been used to avoid overfitting and to generalize the model.

The experimental results shown that the Support Vector Machine with Linear kernel have higher Accuracy among other classifiers. Our methodology produced promising results even for tweets which have been retrieved from the twitter API. The system can be extended to incorporate suitable statistical techniques and further hyperparameter tuning to analyze the classification performance.

REFERENCES

[1] Ashwin Perti, Munesh Chandra Trivedi, Amit Sinha, "Development of intelligent model for twitter sentiment analysis", Materials Today proceedings, issue 1, August,2020.

[2] Gonzalo A.Ruz, Pablo A. Henríquez, Aldo Mascareno, "Sentiment analysis of Twitter data during critical events through Bayesian networks classification", Future Generation Computer Systems ,Volume 106,issue May 2020.

[3] Richard A.Plunz, Yijia Zhoua, Maria Isabel Carrasco Vintimillab, Kathleen Mckeownc, Tao Yud, Laura Uguccionia, Maria Paola Sutto, "Twitter Sentiment in New York city parks as measures of well being", Volume 189, issue April 2019.

[4] Soumya S, Pramod K.V "Sentiment Classification Malayalam tweets using machine learning techniques", ICT Express, issue 22, April 2020.

[5] Akilandeswari Ja, Jothi G, “Sentiment Classification of Tweets with Non-Language Features”, ICACC-2018, Procedia Computer Science 143 (2018) 426–433.
[6] Filippo Chiarello, Andrea Bonaccorsi, Gualtiero Fantoni, “Technical Sentiment Analysis Measuring Advantages and Drawbacks of New Products Using Social Media”, Computers in Industry, Volume 123, December 2020.
[7] Mohd Zeeshan Ansaria , M.B. Aziza, M.O. Siddiquib ,H. Mehraa , K.P.Singha “Analysis of Political Sentiment Orientations on Twitter “, ICCIDS 2019, Procedia Computer Science 167 (2020) 1821–1828.
[8] Sahar A. El_Rahman, Feddah Alhumaidi AlOtaibi, Wejdan Abdullah AlShehri “Sentiment Analysis of Twitter Data”, International Conference on Computer & Information Science (ICCIS) ,2019.
[9] Ankita Sharma, Udayan Ghose, “Sentimental Analysis of Twitter Data with respect to General Elections in India”, ICITETM 2020, Procedia Computer Science 173 (2020) 325–334.
[10] Sushree Das, Ranjan Kumar Behera, , Mukesh kumar, Santanu Kumar Rath, “Real-Time Sentiment Analysis of Twitter Streaming data for Stock Prediction”, ICCIDS 2018, Procedia Computer Science 132 (2018) 956–964.
[11] Dhanush M, Ijaz Nizami S, Abhijit Patra, Pranoy Biswas, Gangadhar Immadi, “Sentiment Analysis of a Topic on Twitter using Tweepy”, IRJET, Volume 5, Issue 05 May-2018.
[12] Erik Cambria, Affective computing and sentiment analysis, IEEE Intell. Syst. 31 (2) (2016) 102–107.
[13] Yequan Wang, et al., Sentiment analysis by capsules, in: Proceedings of the 2018 World Wide Web Conference, 2018.
[14] Ning Liu, et al., Attention-based sentiment reasoner for aspect-based sentiment analysis, Hum-Cent. Comput. Inf. Sci. 9 (1) (2019) 35.
[15] Deepu S. Nair, et al., Sentiment analysis of malayalam film review using machine learning techniques, in 2015 International Conference on Advances in Computing, Communications and Informatics, ICACCI, IEEE, 2015.
[16] S. Soumya, K.V. Pramod, Sentiment analysis of malayalam tweets using different deep neural network models-case study, in 2019 9th International Conference on Advances in Computing and Communication, ICACC, IEEE, 2019.
[17] Alexander Pak, Patrick Paroubek. (2010) “Twitter as a Corpus for Sentiment Analysis and Opinion Mining.” Proceedings of the Seventh conference on International Language Resources and Evaluation LREC'10, European Language Resources Association ELRA. 10: 1320-1326.
[18] Saif, H., He, Y., and Alani, H (2012) “Semantic Sentiment Analysis of Twitter.” International conference on The Semantic Web, Springer Verlag Berlin. 508-524.
[19] Khan, F. H., Bashir, S., & Qamar, U (2014) “TOM: Twitter opinion mining framework using hybrid classification scheme.” Decision Support Systems, 57: 245-257.
[20] Dataset:https://www.kaggle.com/kazanova/sentiment140