INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 5, Issue 2)
Available online at: www.ijariit.com

An efficient approach for sarcasm detection in tweets using polarity flip

Karthika K.
kndkarthi@gmail.com
Institute of Road and Transport Technology,
Erode, Tamil Nadu

ABSTRACT

Sentiment analysis is the qualitative method that uses natural language processing, text analysis, and computational linguistics. The goal of sentiment analysis is to determine if a specific passage in the text shows positive, negative or neutral sentiment towards the subject. Social media platforms, like Twitter, offer a wide ability that allows users to express their thoughts by employing figurative language devices such as sarcasm to achieve different communication purposes. Dealing with such kind of content represents a big challenge for computational linguistics. Due to these difficulties and the inherently tricky nature of sarcasm, it is generally ignored during social network analysis. As a result, the outcome of such analysis is affected adversely. Also, nowadays, Customers use twitter as a platform to express their emotions and opinions about social issues, products or services. Lexicon based methods of text mining can at times fail to recognize sarcasm used by customers online. This has direct implications for companies using text mining on online content to identify new customers, address customer problems, and reduce customer turn rates or for any other CRM (Customer Relationship Management) activities. Thus, sarcasm detection poses to be one of the most critical problems which we need to overcome while trying to yield high accuracy insights from abundantly available data. This work considers sarcastic tweets (including those based on products) and proposes an effective approach to detect sarcasm. The system utilizes 21 features based on context, contrast, and emotions. The results show that sarcastic tweets inclined to have more polarity flips than Non-sarcastic tweets. Also, it is found that MLP and the Random Forest classifier tend to perform better than other classifiers with an accuracy of 94%.

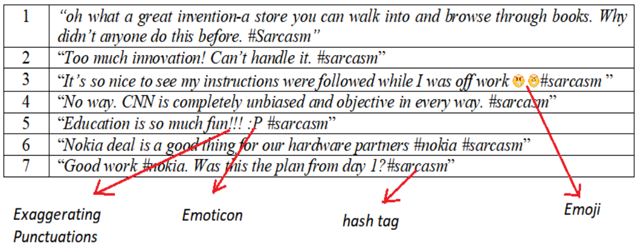
Keywords*— Natural Language Processing, Machine learning, Twitter*

1. INTRODUCTION

Sarcasm is a superior kind of sentiment which shows an interfering factor that can toss the polarity of a given text. The Oxford dictionary defines sarcasm as, “The use of irony to mock or convey contempt” and it is defined by the Free Dictionary [1] as “A form of verbal irony that is intended to express contempt or ridicule”. The fanciful nature of sarcasm makes it a challenge for sentiment analysis.

Sarcasm detection from text has now protracted to different data forms and methods. This Collaboration has resulted in motivating novelties for automatic sarcasm detection. To predict the correct sentiment of any text, identifying the sarcasm is vital. The challenges of sarcasm and the advantage of sarcasm detection to sentiment analysis have paved way to an interest in automatic sarcasm detection as research work. Automatic sarcasm detection refers to an approach that computationally predicts if a given text is sarcastic or not. For instance, the sentence “I love it when you stare at me!!” should be predicted as sarcastic, while the sentence “I love it when you welcome me with a bouquet!!” should be predicted as non-sarcastic. Though this can be manually predicted easily, for automatic sarcasm detection, it’s a big challenge. The difficulty is due to the nuanced ways in which sarcasm may be expressed. Recognizing sarcasm is one of the truly obscure tasks in Natural Language Processing (NLP) too.

Twitter, one of the largest web source of the era for grabbing opinions and a perfect destine for the people to share their thoughts, spark a global conversation and convey real-time events. As on October 2017, it claims to possess 330 million active users around the globe. The voluminous data generated from such a data ocean has been cherished by many companies and organization for improving their business. According to its official marketing site [2], Twitter states that it offers new marketing mix modeling insights and guidance on where to invest. It also utters that Twitter marketing campaigns can deliver 40% higher return on investment compared to other media channels. It assists brand marketers in measuring the impact of their advertisements. Besides that, it shares the keys to In-Stream Video Ad effectiveness. The evolving video consumption patterns represent a new opportunity for marketers. Twitter’s new video ad solution performs as well as, and in some cases better than, a leading video platform. However, mining valuable insights from twitter is not a straight forward task due to its noisy nature in terms of the tweets being posted. Adding to that sarcasm also poses to be a great challenge for deriving accurate results of the analysis. Figure 1 shows a few sample tweets that are sarcastic.



1	*"oh what a great invention-a store you can walk into and browse through books. Why didn't anyone do this before. #Sarcasm"*
2	*"Too much innovation! Can't handle it. #sarcasm"*
3	*"It's so nice to see my instructions were followed while I was off work #sarcasm "*
4	*"No way. CNN is completely unbiased and objective in every way. #sarcasm"*
5	*"Education is so much fun!!! :P #sarcasm"*
6	*"Nokia deal is a good thing for our hardware partners #nokia #sarcasm"*
7	*"Good work #nokia. Was this the plan from day 1?#sarcasm"*

Fig. 1: Sample sarcastic tweets

2. RELATED WORKS

In the last few years, more attention has been given to twitter sentiment analysis by researchers, and a number of recent papers have been addressed to the classification of tweets. However, the nature of the classification and the features used vary depending on the aim. Researchers have used different features like lexical feature (uni-gram, bi-gram, tri-gram and n-gram through which one can easily identify sarcasm in text), Pragmatic feature (emoticons, smiles, replies, etc.), Intensifiers (Adverbs, adjectives, etc.), interjection words (wow, oh, uh etc.) and high frequency words.

The contributions of a few authors like Joshi et al., Riloff et.al, Tomoaki Ohtsuki et al. etc.., are inevitable when it comes to linguistics based analysis. Most recent works show that deep learning based models outperform machine learning based models as they use heterogeneous features to detect sarcasm.

Le Hoang Son et al. [6] propose a deep learning model called sAtt-BLSTM convNet that is based on the hybrid of soft attention-based bidirectional long short-term memory (sAtt-BLSTM) and convolution neural network (convNet) applying global vectors for word representation (GLoVe) for building semantic word embedding's. In addition to the feature maps generated by the sAtt-BLSTM, punctuation-based auxiliary features were also merged into the convNet. The training- and test-set accuracy metrics was performed to compare the proposed deep neural model with convNet, LSTM, and bidirectional LSTM with/without attention and it was observed that the novel sAtt-BLSTM convNet model outperforms others. However, the system could have enhanced with novelty in vocabulary with advanced architectures.

Another interesting recent work was [5] in which sarcasm interpretation was performed as converting a sarcastic utterance into its non-sarcastic (literal) interpretation. The system presented three approaches: (a) a rule-based approach that considers sarcasm as a form of dropped negation and associate negation words with verbs present in the sarcastic utterance, (b) Statistical Machine Translation-based (SMT) approach that addresses the sarcasm interpretation task as monolingual machine translation and (c) three Deep Learning-based (DL) architectures, Encoder-Decoder Network, Attention Network, and Pointer Generator Network. BLEU score, METEOR score, and ROUGE-L score were used for system evaluation. It was found that Encoder-Decoder network obtains the best ROUGE-L score of 80.89. However, for more thorough evaluation other metrics such as word error rate and human level evaluation could have been considered. However, the system could show still better performance if a larger parallel corpora of sarcastic tweets were considered.

Hence the main goal of the proposed approach is to provide a better combination of techniques that would enhance the accuracy of the text-based analysis.

This section should be succinct, with no subheadings. This heading should be Times New Roman 10-point boldface, initially capitalized, flush left, with one blank line before.

Recently a system proposed by Sree lakshmi K et. al [3] was based on Joshi et al [4] that focuses on context incongruity. It pointed out two major kinds of context incongruities present in sarcastic tweets as Explicit and Implicit. They highlighted that either one of them is present in more than 70% of tweets. Explicit incongruity is expressed by directly specifying sentiment words of both polarities, whereas implicit incongruity is expressed through phrases of implied sentiment, as opposed to using polar words. They modeled a classifier with SVM and RBF kernel that had Joshi et al [4], as a baseline. It was projected that the accuracy of the model improved by 2.6% and 2.8% for 5-fold and 10-fold cross validation respectively. However, it failed to consider the inter-sentential incongruity.

A novel self-deprecating sarcasm detection approach using an amalgamation of rule-based and machine learning techniques were dealt in [9]. The rule-based techniques were used to identify candidate self-around tweets, whereas machine learning techniques were used for feature extraction and classification. It included six self-deprecating features and five hyperbolic features are identified to train three different classifiers – decision tree, Naive Bayes, and bagging.

The objective of the system proposed by Anukarsh et al. [7] is to counter the problems faced in the social media dataset. The authors claimed that sarcastic tweets can mislead data mining activities and result in the wrong classification. This paper compares various classification algorithms such as Random Forest, Gradient Boosting, Decision Tree, Adaptive Boost, Logistic Regression and Gaussian Naïve Bayes to detect sarcasm in tweets from the Twitter Streaming API. The best classifier was chosen and paired with various pre-processing and filtering techniques using emoji and slang dictionary mapping to provide the best possible accuracy. The emoji and slang dictionary being the novel idea introduced in this paper. For evaluating the proposal, it tabulated and showed the results of testing for a split of 60:40, 70:30 and 80:20 with and without the emoji and slang dictionaries for 6 classifiers (Random forest, Gradient Boost, Decision tree, AdaBoost, Logistic regression, Gaussian Naïve Bayes). The table showed that the best performing algorithm was the Gradient Boosting algorithm which optimizes the cost function by considering the weak hypothesis that will make the classifier to classify wrongly.

The unaddressed part was that the punctuation related features were not considered during feature selection. The model could have been prepared to capture live streaming tweets by filtering through hashtags and then perform immediate classification.

Tomoaki Ohtsuki et al. [8] claimed that recognizing sarcastic statements can be very useful to improve automatic sentiment analysis of data collected from microblogging websites or social networks. It proposed a pattern-based approach to detect sarcasm on Twitter. The authors proposed four sets of features that cover the different types of sarcasm and then they used those to classify tweets as sarcastic and non-sarcastic. In particular, they emphasized the importance of pattern-based features for the detection of sarcastic statements.

To measure the potential of their method, they considered the approach proposed by Riloff et al. [10] as well as the n-gram-based approaches as their baseline. The accuracy, precision, recall, and F-score of the proposed method was higher than that of the baseline. The unaddressed part was that it did not consider other domains like movie or product reviews.

An automatic sarcasm detection was employed in [12] to categorize the tweets and product review texts. Similarly, a scoring system based on sarcasm for sentiment analysis was used in [11]. It mainly insisted that different people have a different interpretation of sarcastic texts and that it leads to debatable opinions about the product which is being considered. Hence the authors claimed that recognizing sarcastic statements can be very useful as it enhances the efficiency of after-sales services or consumer assistance through understanding the intentions and real opinions of consumers when browsing their feedbacks or complaints. It proposed a system that will measure sarcasm using tweets from Twitter. It used a different algorithm to calculate the effect of sarcasm on texts and generate a score. Different features were generated from the received tweets which in turn helped them to generate the score.

They compared the scores from different algorithms to present the most efficient way to detect sarcasm. The system also provided a separate portal to check the score of any sentence/text entered by the user and determine its score using the most accurate algorithm. To evaluate their method Precision, Recall and F score were used to compare the Linear SVC, Logistic regression, Naive Bayes algorithms for identifying the one that gives accurate scores for sarcastic and non-sarcastic reviews. But for training, the model, only Linear SVC, Logistic regression, and Naive Bayes algorithms were considered. Multivariate regression or multiple regression or Lasso regression could have also been used for the same purpose.

3. PROPOSED SYSTEM

Figure 2 shows the general block diagram of the proposed system. It involves Extracting tweets, Preprocessing, Getting the emoticon and slang dictionary ready, Extracting features, Identifying the polarity& sarcasm, Training the model for Sentiment Classification. Finally validating the enhancement inaccuracy that is being proposed by the model.

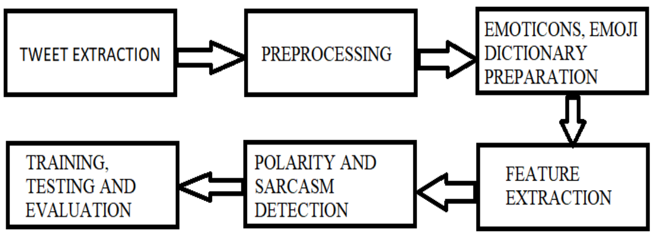


Fig. 2: General block diagram

3.1 Tweets extraction

In Twitter, an application can extract data by using APIs that include search API, REST API, and streaming API. Tweepy Python library was used for accessing the Twitter API. OAuthHandler () and set_access_token () were used to authenticate and set up access to the streaming Twitter data.

3.2 Preprocessing

Preprocessing is one of the most important steps of Text Mining. For accurate information retrieval, we should have proper data set with accurate features. In text preprocessing, there are many steps like removing stop words, URLs, stemming and finally converting it to a uniform format.

The NLTK (natural language toolkit) was used for preprocessing. Few of the basic functions include, (a) word_tokenize (for segmenting running text into words) (b) Porter Stemmers (that remove morphological affixes from words, leaving only the word stem) (c) WordNetLemmatizer (that determines the lemma of a word based on its intended meaning) (d) Stop word removal using the NLTK corpus.

3.3 Dictionary preparation

It involves the collection of the emoji and emoticons along with their corresponding sentiment score.

3.4 Feature extraction

Feature extraction refers to the extraction of linguistic items from the documents to provide a representative sample of their content. Distinctive vocabulary items found in a document are assigned to the different categories by measuring the importance of those items to the document content. It is the process of transforming the input data into a set of features which can very well represent the input data. Here 21 features were considered, the list of which is given in table 1. The correlation matrix of the features is given in figure 2. Sentiment Intensity Analyzer functionality of NLTK was used for quantitative analysis of the tweets.

Table 1: Features list and their description

Feature	Description
User_mention_count	The number of user mentions present in the given tweet.
Exclamation_count	The number of Exclamation Marks in the tweet.
Questionmark_count	The number of Question marks in the tweet.
Ellipsis_count	The number of '…' ellipsis in the tweet
interjection_count	Counts the interjections in the tweet.
Uppercase count	Counts the capital words in the tweet.
RepeatLetter_count	Counts the repeated letter in a word (ex. "Whaaat")
sentiment score	Sentiment score of the tweet.
Positive_word_count	The number of positive sentiment words
Negative_word_count	The number of negative sentiment words
PolarityFlip_count	The number of polarity flip in a tweet i.e positive to negative or negative to positive change.
Noun_count	The number of Nouns in a tweet.
Verb_count	The number of Verbs in a tweet.
Positive_intensifier_count	The number of positive intensifiers
Negative_intensifier_count	The number of negative intensifiers
skip_bigrams_sentiment	The sentiment of each bigrams in the tweet.
skip_trigrams_sentiment	The sentiment of each trigrams in the tweet.
skip_grams_sentiment	The sentiment of each bigrams in the tweet.
emoji_sentiment	Calculate the emoji sentiment from the emojis present in the tweet.
Passive_aggressive_count	The total number of passive aggressive statements in a tweet.
emoji_tweet_flip	Indicates if there is any contradiction between the sentiment score of the text and the sentiment score of the emoji.
hashtag_sentiment_score	Calculates the polarity of the hashtag.

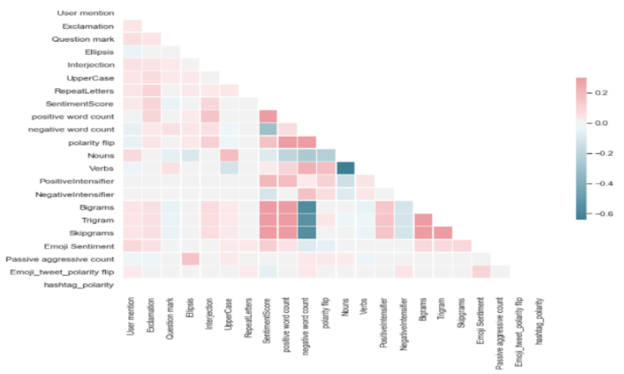


Fig. 3: Correlation matrix of the features

3.5 Polarity and sarcasm detection

Polarity Detection involves identifying the sentiment orientation of a given sentence. The sentiment orientation of the word can be changed from positive to negative or vice-versa. Usually, the flipping in polarity is termed as "polarity Shift" or "Valence Shift". It was identified by comparing the polarity scores of each of the tokens Sarcasm detection is the process of identifying and classifying the tweets as sarcastic or not. The classification is done based on the extracted features.

3.6 Training, testing, and evaluation

For training the Classification: Random Forest, Support Vector Machine, Gradient Boost, AdaBoosting, Stochastic Gradient Descent, MLP (Multilayer Perceptron), Naïve Bayesian, Decision Tree etc. were used. Sklearn python library that built upon the SciPy (Scientific Python), was used for the machine learning algorithms. 10-Fold cross-validation was used to evaluate the predictive model by partitioning the original sample into a training set and a test set. To compare the results three parameters were considered, namely, precision (Equation 1), recall (Equation 2) and f score (Equation 3).

Precision is a statistical parameter that shows, how many tweets correctly identified out of total identified tweets by an algorithm with the reference of ground truth. Correctly identified tweets are called true positive (Tp) as algorithm and ground truth, both identified positive and the tweets are incorrectly identified called false positive (Fp) as algorithm identified positive but ground truth was negative. Similarly, how many of tweets correctly rejected is called true negative (Tn) as both algorithm and ground truth identified negative and how many tweets incorrectly rejected called false negative (Fn) as algorithm identified negative but ground truth was positive.

$$Precision = \frac{T_p}{T_p + F_p} \tag{1}$$

$$Recall = \frac{T_p}{T_p + F_n} \tag{2}$$

$$F - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

Where, Tp = True positive, Fp = False positive, Fn = False negative.

4. RESULTS AND DISCUSSION

The experimental setup was done in Windows, Anaconda3 environment, python version 3.6.6 and in Tableau for visualizations. For ease of application, Jupyter notebook was used. A Twitter application was created to generate keys and tokens. With them, the authentication was done to extract tweets in Python. To collect sarcastic tweets, "#sarcasm" and "#not" hashtags were used. To collect non-sarcastic tweets, general product based tags were used. It included "#nokia", "#iphone" and "#blackberry". Only English tweets were extracted for this experiment. Also, the extraction was done after filtering out the retweets. 91298 tweets were extracted out of which 51300 were sarcastic and 39998 were non-sarcastic. This dataset was preprocessed and used for feature extraction. The following values were taken: User mention count, Exclamation mark count, Question mark count, Ellipsis count, Interjection count, Upper Case count, Repeat Letters count, Sentiment Score, positive word count, negative word count, polarity flip, Nouns count, Verbs count, Positive Intensifier count, Negative Intensifier count, skip Bigrams sentiment, skip Trigram sentiment, Skip grams sentiment, Emoji Sentiment, Passive aggressive count, Emoji tweet polarity flip, hashtag sentiment score. Based on the specified features listed above a CSV file for the feature values was created.

The following classifiers were considered separately: SVM, K nearest neighbor, Logistic Regression, Decision Tree, Naïve Bayesian, Multilayer Perceptron (MLP), AdaBoosting ensemble, Gradient Boosting, Random Forest, Stochastic Gradient Descent, Linear Discriminant Analysis (LDA). 10-Fold cross-validation was used for the predictive model. The results of the evaluation metrics: Precision, Recall, F1-Score, and accuracy, for each of the classifier is shown the table 2.

Table 2: Results of the Metrics for each of the classifiers

CLASSIFIER	Precision	Recall	F1-score	Accuracy
SVM	0.9	0.9	0.9	0.90
k-neighbor	0.91	0.91	0.91	0.91
Logistic Regression	0.66	0.65	0.63	0.64
Decision Tree	0.92	0.92	0.92	0.91
Naïve Bayesian	0.82	0.81	0.81	0.81
Neural Networks (MLP)	**0.94**	**0.94**	**0.94**	**0.94**
AdaBoosting	0.93	0.92	0.92	0.92
GradientBoosting	0.93	0.93	0.93	0.93
Random Forest	**0.94**	**0.94**	**0.94**	**0.94**
LDA	0.9	0.9	0.9	0.89
Stochastic gradient Descent	0.74	0.74	0.73	0.74

The experimental results show that Random forest and Multi-Layer Perceptron outperform the other classifiers. Also by comparing the values of the features as shown in figure 3, it can be seen that the values of certain features namely: sentiment score, verb count, positive word count, negative word count, polarity flip, count of question marks and exclamation mark, count of positive intensifier and negative intensifiers, count of interjections, count of repeat letters were higher in sarcastic tweets than in non-sarcastic tweets. The highlighted comparison of polarity flip in sarcastic and non-sarcastic tweets is shown in figure 4.

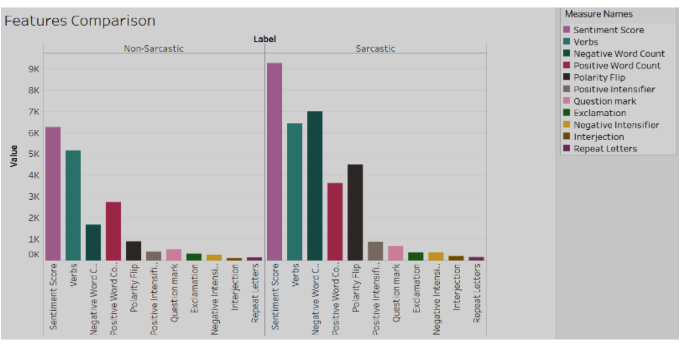


Fig. 4: Cumulative features value comparison

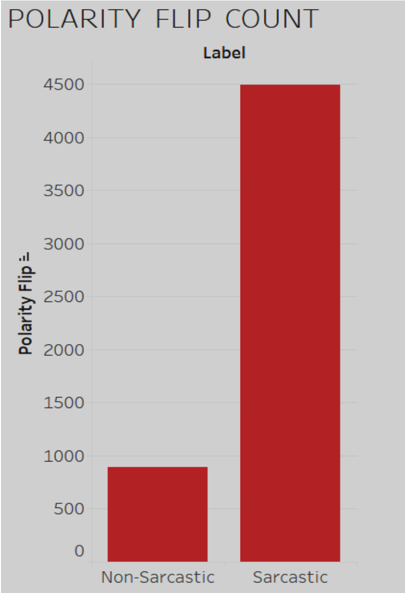


Fig. 5: Polarity flip count comparison

5. CONCLUSIONS AND FUTURE WORKS

Sarcasm analysis is one of the most important and challenging tasks as it doesn't have any pre-defined structure. Interest in detecting the presence of sarcasm in social media texts has grown significantly in recent years. Also for the areas related to sentiment analysis, where it is important to avoid misinterpreting sarcastic statements as literal. As far as the business applications are concerned, analysis done after handling sarcastic texts are more purposeful and yields an accurate classification of sentiments, from which the right informed decisions can be made. Hence the proposed system provides a way of improving the existent sarcasm detection by considering the importance of polarity flipping. This work derived analytical views from a social media dataset i.e., twitter dataset. The proposed method makes use of 21 different features, which directly contribute to the improvement of accuracy. Also from the comparison of a set of classifiers used on these features, MLP classifier, and Random Forest seems to outperform with an accuracy of 94%.In the future, Deep learning based methods can be used to improve the accuracy in detecting sarcasm which in turn enhance the performances of sentiment analysis. Moreover, Sarcasm detection in non-"#sarcasm" hashtag tweets, is much more challenging than those tweets with "#sarcasm" hashtag.

This should clearly explain the main conclusions of the work highlighting its importance and relevance.

6. REFERENCES

[1] https://www.thefreedictionary.com/
[2] https://marketing.twitter.com/na/en/insights.html
[3] Sreelakshmi K, Rafeeque P C, "An Effective Approach for Detection of Sarcasm in Tweets", International CET Conference on Control, Communication, and Computing (IC4), pp.377 - 382,5–7 July 2018.
[4] A. Joshi, V. Sharma, and P. Bhattacharyya, "Harnessing context incongruity for sarcasm detection.," ACL (2), pp. 757–762, 2015.
[5] Abhijeet Dubey, Aditya Joshi, Pushpak Bhattacharyya, "Deep Models for Converting Sarcastic Utterances into their Non-Sarcastic Interpretation", 6th ACM IKDD CoDS and 24th COMAD (CoDS-COMAD'19), January 3–5, 2019, Article 4,4 pages https://doi.org/10.1145/3297001.3297043
[6] Le Hoang Son, Akshi Kumar, Saurabh Raj Sangwan, Anshika Arora, Anand Nayyar, Mohamed Abdel-Basset, "Sarcasm Detection Using Soft Attention-Based Bidirectional Long Short-Term Memory Model with Convolution Network", IEEE Access (Early Access), pp.1, February 2019.
[7] Anukarsh G Prasad; Sanjana S, Skanda M Bhat, B S Harish, "Sentiment Analysis for Sarcasm Detection on Streaming Short Text Data", 2nd International Conference on Knowledge Engineering and Applications IEEE, pp.1-5, 21-23 October. 2017.
[8] Mondher Bouazizi, Tomoaki Ohtsuki, "A Pattern-Based Approach for Sarcasm Detection on Twitter", IEEE Access, pp. 5477 – 5488, 24 August 2016.
[9] Namrata Bhan, Prof. Mitchell D'silva, "Sarcasmometer using Sentiment Analysis and Topic Modeling", International Conference on Advances in Computing, Communication and Control (ICAC3), pp.1-7, 1-2 December. 2017.
[10] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, "Sarcasm as the contrast between a positive sentiment and negative situation", in Proc. Conf. Empirical Methods Natural Language Processing, pp. 704–714, October. 2013.
[11] Muhammad Abulaish, Ashraf Kamal, "Self-Deprecating Sarcasm Detection: An Amalgamation of Rule-Based and Machine Learning Approach", IEEE/WIC/ACM International Conference on Web Intelligence (WI), pp. 574 – 579, December 2018.
[12] Paras Dharwal, Tanupriya Choudhury, Rajat Mittal, Praveen Kumar, "Automatic sarcasm detection using feature selection", 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), pp. 29–34, December 2017.

