




Sentiment analysis of sarcasm detection in social media

Sunusi Kabir Alaramma¹, Adamu Adamu Habu², Badamasi Imamu Ya'u¹, and Abdullahi Gamsha Madaki¹

¹Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi, Nigeria.

²Department of Computer Sciences, University of St Andrews, Jack Cole Building, North Haugh, St Adrews, KY16 9SX, UK.

*Correspondence: skalarammane@gmail.com; aah5@st-andrews.ac.uk; +234813858559

Abstract	Article History
<p>When people talk or write more especially on social media, they tend to sometimes, flip the polarity of their expressions. Oftentimes, this change of mood affects the accuracy of the sentiment analysis. Sarcasm is a sophisticated form of irony widely used in social networks and micro blogging websites and is usually a type of sentiment where person speaks the contradictory of what individual word means, expressing gloomy feelings applying positive words. However, it is hard even for humans to recognize. Hence, sarcastic expressions play a vital role in the outcome of automatic sentiment detection in sentiment analysis. To improve on the performance of automatic detection of sentiments in social media networks, there is need for sarcastic statements to be recognised and classify correctly. Sentiment Analysis is a method for identifying people's opinion, attitude, sentiment, and emotion towards any specific target such as individuals, events, topics, product, organizations, services etc. This paper proposed an ensemble classification method having base classifiers as Random Forest, K-Nearest Neighbour and Naive Bayes for detecting sarcasm using lexical, pragmatic, linguistic incongruity and context incongruity features. The same ensemble method and Support Vector Machines (SVM) were used to identify the sarcasm types so as to diagnose the mood of the users. We report the results and present a comparative evaluation of the ensemble method and SVM classifiers for sarcasm and sarcasm type detection indicating their suitability for the task. The results obtained from the set of experiments conducted indicate the reliability of our approach.</p>	<p>Received: 21/03/2023 Accepted: 18/05/2023 Published: 06/06/2023</p> <p>Keywords Sarcasm; Twitter; Neural Network; Machine Learning; Sentiment Analysis</p> <p>License: CC BY 4.0*</p>  <p>Open Access Article</p>
<p>How to cite this paper: Alaramma, S.K., Habu, A.A., Ya'u, B.I., and Madaki, A.G. (2023). Sentiment analysis of sarcasm detection in social media. <i>Gadau J Pure Alli Sci</i>, 2(1): 76-82. https://doi.org/10.54117/gjpas.v2i1.72.</p>	

1.0 Introduction

Sarcasm is a phenomenon in which the literal meaning of the text or speech is different from the implied sentiments. Sarcasm is speech or writing actually means the opposite of what it seems to say. For example, if someone says, "Woww! He's cleverer than Shehu Jaha!" sarcastically, it conveys that the target said something simple or obvious and the speaker is mocking him. Sarcasm is used not only to make fun but also for criticizing other people, views, ideas etc. due to which sarcasm is very much used on twitter (Aditya *et al.*, 2016).

The inherently ambiguous nature of sarcasm sometimes makes it hard even for humans to decide without prior knowledge of the topic. Context and the tone in which syllables are stressed play a vital role in detecting sarcasm in sentences. But tone is not commonly indicated in written form in a lot of languages and as such sometimes, sarcasms in written texts are missed altogether (Anukarsh, *et al.*, 2017). Numerous studies however, attempted to solve this by introducing different punctuation marks to indicate sarcasm but the success is not much. In a verbal dialogue however, presence of certain verbal or physical cues, help in easily identifying sarcasm

(Davidov *et al.*, 2010). Similarly, numerous standard datasets were developed and used for sarcasm detection tasks which include SemEval 2018 Task 3 as used in Shrivastava and Shishir (2021) SARC in Du *et al.* (2022), and Internet Argument Corpus (IAC) used in Ren *et al.* (2020).

Interest in Natural Language Processing (NLP) research has grown manifolds in the last decade or so and most of it is centred on analysis of social media text to solve problems like sentiment analysis etc. (Liebrecht and Kunneman, 2013). Ensemble approach is a promising technique in machine learning obtain a greater predictive ability by aggregating the output of the base classifiers into a single combined output than its comprising algorithms (Bamman and Smith, 2015). To obtain M different models of classification, they applied M different machine learning algorithms on the same dataset. An alternative approach is to build M datasets from the training data and use single learning algorithm for each dataset (Ashwin *et al.*, 2015).

Work done by Anukarsh, *et al.*, (2017) proposed a methodology to detect sarcastic and non-sarcastic tweets based on the slang and emojis used in their tweets. They considered the values for slang and emoji used from the slang dictionary and emoji dictionary. Then these values are compared with different classification algorithms like Random Forest, Gradient Boosting, Adaptive Boost, Gaussian Naive Bayes, Logistic Regression, and Decision Tree, to identify the sarcasm in tweets from the Twitter Streaming API. From all these classification algorithms considered, the best is identified and combined with different pre-processing and filtering techniques using emoji and slang dictionary mapping to yield the finest efficiency. Manoj and Pallavi (2017) proposed a new approach for sarcasm detection as NLP and corpus-based approach. The objective was to identify the intention to use the sarcastic statement in the tweets by individuals. Tweets from the Twitter were collected and NLP techniques like tokenization, PoS, and lemmatization were performed. NLP techniques on tweets are applied to fetch action words. Once the action words are found from the tweets, these are matched with the corpus of sarcasm data using semantic matching and graph-based matching which gives a score of sarcasm for the given tweet. By this score, the level of sarcasm in the given tweet is detected.

Aditya *et al.* (2016) proposed new approaches for automatic sarcasm detection and also detected three milestones in history of sarcasm detection research. He used semi-supervised pattern mining to identify the implied sentiment and also used hashtag based supervision and proposed the use of context beyond target text. Rule based method was used in order to capture evidence of sarcasm in the text.

Isidoros *et al.* (2016) proposed a sentiment analysis method for automated emotion detection in a text using classifier ensembles. The ensemble classifier was based on three key classifiers—a learner from Naive Bayes, a maximum entropy and knowledge-based system. Results showed that ensemble schema performed better in recognizing emotion present in the text and determining text polarity than the sole classifiers.

Paras *et al.* (2017) suggested in his work that it was a very complex task to detect sarcasm in automatic manner in Sentiment Analysis. The detection task included composite linguistic analysis and machine learning algorithms. In this work, a lot of sarcasm analysis techniques were reviewed. These techniques were used to filter the sarcastic statements from a text. In the work of Santosh *et al.* (2017) several sarcasm detection approaches had been recommended for detecting sarcasm within text. A novel context-based pattern had been recommended in this work to detect sarcasm within Hindi tweets. The suggested technique using ensemble method recorded an accuracy of 87% when used on Hindi news dataset.

Shubhadeep *et al.* (2017) considered different text independent feature sets. This kind of features included n-grams and function words. In this work, sarcasm was detected by adding other features that depicts writing styles and an accuracy of about 65% was recorded using Naive Bayes and fuzzy clustering algorithms.

Collecting enough sarcastic data to train a model for sarcasm detection is itself an unsurmountable task. From 2006, Twitter has become the single most valued tool across the entire range of NLP tasks. Authors in Anukarsh *et al.* (2017) and Manoj *et al.* (2017) proposed using tweets author annotated with #sarcasm or #irony for accumulation of sarcastic text. Authors in Anukarsh *et al.* (2017) make use of unigram, bigram and trigram features for identifying sarcasm from text using Balanced Winnov and achieve an accuracy of 75%. Relying on the insights of authors in Joshi (2016), most sarcasm detection approaches treat the task primarily as a text categorization problem, and use lexical and linguistic features such as interjections, intensifiers, non-veridicality and hyperbole, that is, three positive or negative words in a row to do the detection job.

In Pomima *et al.* (2017), the authors used Naive Bayes classifier and SVM, for detecting sarcasm in Indonesian Social media text, using features like negativity and number of interjections. They used negativity to capture the global sentiment value and interjection to exemplify the lexical singularities in the typescript. They discover that negativity features are not really beneficial as a large number of sarcastic texts have no global topic, and marking the text topic is not recognized. Authors in Prasad *et al.* (2017) also

suggest the use of pattern based and punctuation based features with Support Vector Machines for sarcasm detection.

Mondher *et al.* (2017) proposed an innovative method based on pattern for twitter sarcastic text detection. In this work, four feature sets had been utilized. These feature sets covered the different sorts of identified sarcasm and their significance was also considered. These feature sets were utilized for classifying tweets based on their sarcastic polarity. The model developed has achieved an accuracy of 83.1% and precision rate of 91.1% using an ensemble method.

The work of Bhakuni *et al.* (2021) has recorded a good result by analysing and comparing a number of machine learning algorithms including SVM, decision tree, K-Nearest Neighbour and Naïve Bayes in detecting sarcasm using data from Twitter. The result show that the classifiers have accuracy of 93%, 83%, 86% and 65% for SVM, Naïve Bayes, Decision Tree and K-Nearest Neighbour respectively.

Using SVM, K-Nearest Neighbor, Linear Discriminant Analysis, Decision Tree and Logistic Regression were used in the research conducted by Razali *et al.* (2021) to detect sarcasm using deep features extracted using Convolutional Neural Network (CNN). The findings show that Linear Regression model has the highest accuracy of 89%.

Apart from Machine Learning approaches, Deep Learning approaches were also explored in sarcasm detection tasks and several researches were conducted to ascertain their performance on the task. For instance, Ortega-Bueno *et al.* (2019) proposed a model for sarcasm type detection and used POS tags to identify the level of sarcasm in text and the contribution of each to the detection task was also evaluated. The results of the work have been quite remarkable with an accuracy of 83.1% and precision of 91.1%.

Kumar *et al.* (2020) used a dataset with over 20,000 threads from Reddit to detect sarcasm in text using contextual features with three models comprising of Ensemble voting model developed with three base classifiers for training. The second model combines pragmatic and semantic features using TF-IDF and five classifiers using Long Short Term Memory (LSTM). The final model was used to comprehend context and create semantic word embedding. The finding has recorded an accuracy of 91.32%.

The use of user based context to improve the classification has been deployed in work done by Wu *et al.* (2021) with MUsTARD dataset and used neural network for the classification. Their finding proposed the use of incongruity aware network for detecting sarcasm in text.

Buruah *et al.* (2020), Srivastava *et al.* (2020) Geng *et al.* (2022) used data obtained from Twitter and Reddit with different deep learning approaches. Bruah *et al.*

(2020) obtained result with F-Score of 0.74 and 0.66 for the Twitter and Reddit datasets respectively using LSTM while Srivastava *et al.* (2020) achieved F-Score of 0.74 for Twitter dataset and 0.64 for the Reddit using hierarchical BERT. Geng *et al.* (2022) has an accuracy of 87.5% for Twitter dataset when multihead self-attention with BiLSTM was used.

The literatures studied so far have revealed that research works have been conducted to detect sarcasm on social media using diverse datasets that have varied features and as such, have different level of accuracy based on the sentiment analysis approach used. Even with the level of accuracy recorded in the previous researches, there is need for improvement. Choosing the features with the highest contribution and enhancing the underlying classifiers in the ensemble methods is a task that will improve the detection of sarcasm. There is also the need to use detect the different type of sarcasm emotion employed to improve the task.

This study attempted to improve on the detection of sarcasm in social media using some unique textual features that are crucial in the detection task. The proposed scheme will utilise SVM and ensemble method having Random Forest, Naive Bayes and K-Nearest Neighbour as base classifiers for better performance in detecting both the presence of sarcasm and its type.

2.0 Material and methods

The proposed system recognizes the sarcastic emotions of the individuals with the use of both SVM algorithm and an ensemble method and also identifies the type of sarcastic emotions using SVM method and the generated ensemble. The proposed system architecture is shown in the Figure 1. In this work, we first start by obtaining data from Twitter. In the extracted data, we perform preprocessing including feature extraction and selection. Finally, sarcasm detection and identification of categories of sarcasm (manic, depression, polite, brooding) have been performed on the dataset using SVM and ensemble classifiers.

2.1 Data Collection

Twitter is a micro blogging network that allows its users to write tweets of size up to 140 characters. The Twitter official website was used to obtain the dataset needed for the experiments using appropriate APIs and programs in the form of tweets. Tweets of #sarcasm or #sarcastic or #not of a certain length are taken as part of the sarcastic datasets. The API and programs also extract an appropriate number of regular tweets that have positive or negative sentiment scores. These corpora would then be used at a later stage to train and test the systems.

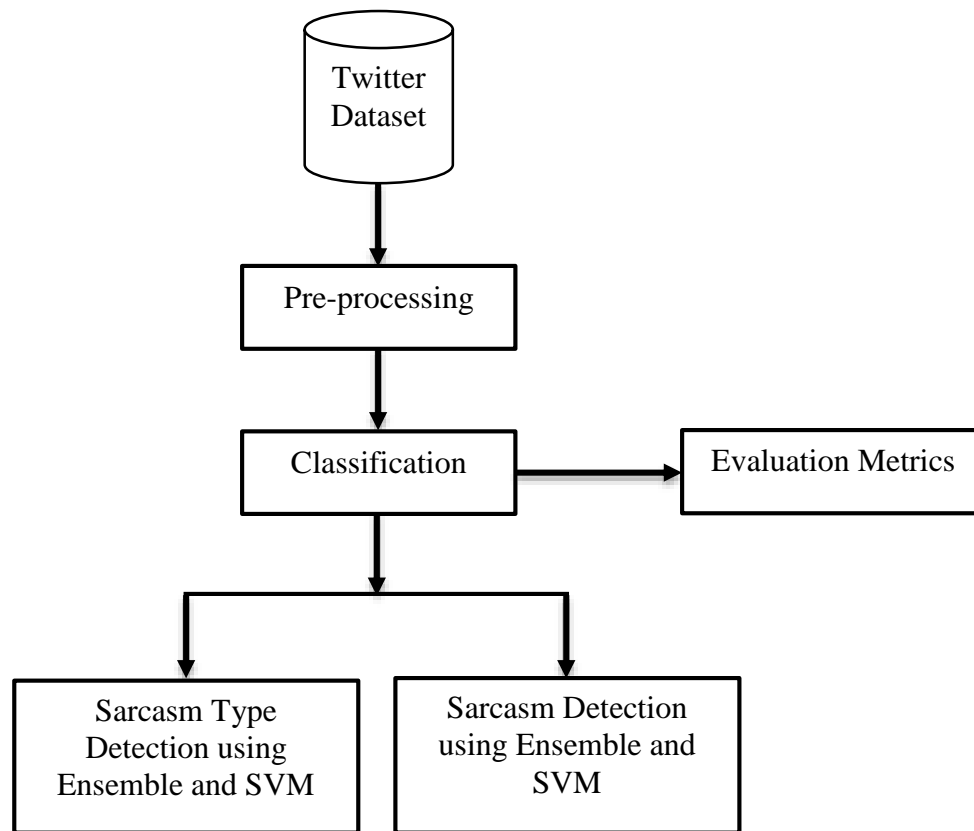


Figure 1. Proposed system architecture

2.2 Data Pre-processing

The collected dataset is then pre-processed to be converted into computer readable format. Following are the steps performed in order to pre-process the data:

- Tokenization: This part of preprocessing produces tokens which are useful elements obtained from broken down of words to symbols, or other elements.
- Data Cleaning: Oftentimes, data obtained from websites come with undesirable elements that do not add to any sentiment in the text and therefore need to be removed. Such elements and metadata including space, hashtag, quotations, retweets, emoticons, and URLs were all detached. Alphabets are converted to lower case and numbers removed. Tweets with very scanty words (less than three words) and meaningless tweets were also deleted.
- Removal of stop words: All stops words are deleted. Stop words are words such as in, which, that etc and they do not add any importance to the sentiment analysis.
- POS (Parts of Speech) tagging: Certain words have greater impact on the classification of

opinions. Words such as noun, adjectives are extracted in this stage and tagged.

- Lemmatization and Stemming: All words have their roots and stemming reduces each word to its root called the stem. Lemmatization adds the character that is missing to the word's root. E.g kicked and kicking produces kick when stemmed.

2.3 Classification

In the proposed methodology, ensemble classifier will be created by selecting Naive Bayes, Random Forest and K-Nearest Neighbour as baseline classifiers.

- Naive Bayes is a supervised machine learning algorithm. It is a probabilistic classifier used successfully in a wide variety of tasks. Naive Bayes is based on Bayes' theorem is used in calculating conditional probability with the following formula:

$$P(A|B) = \frac{(P(B|A) * P(A))}{P(B)} \quad (1)$$

The formula above translates to how often A happens given that B happens, written $P(A|B)$. When we know how often B happens given that A happens written $P(B|A)$ and how

likely A is on its own P(A) and how likely B is on its own P(B). P(A|B) is called posterior probability. The beauty of the classifier is that it needs less number of training data and that it has the basic assumption that each feature makes an independent and equal contribution to the outcome.

- Random Forest as a machine learning algorithm is widely used in both classification and regression tasks in which every leaf node represents a class label. Generally, Random forest builds decision trees on different samples with the internal non leaf node used to test feature and branches for result (Manoj *et al.*, 2017).
- K-Nearest Neighbors also known as KNN is a non-parametric supervised learning algorithm. In both classification and regression tasks that the algorithm is being used for, the input consists of the k closest training examples dataset and tries to determine what group a data point belongs to by looking at the data point around it. It's a simple model that doesn't require much tuning and can handle multiclass problem.

To build the ensemble classifier, we used a combination of the base classifiers using stacking approach. A model was built by combining the decisions of base classifiers through majority voting technique. Also, Support Vector Machines algorithm was used alongside for all the classification tasks.

- *Support Vector Machines (SVM)* is an effective traditional machine learning algorithm for both clustering, regression and classification. The support vector which the algorithm is built upon are data points that are closer to the hyperplane and influence the position and the orientation of the hyperplane. SVM shows very good performance and higher accuracy in many

studies directed towards sentiment analysis in many languages. The work reported in Kreuz (2007) shows that SVM did well with the English language when compared to other classifiers. Some of the features of SVM that makes it very useful in opinion mining includes the good memory efficiency and versatility provided by the kernel function.

3.0 Results and discussion

Twitter, a social media website was used to collect the dataset used in this study using the crawler described in Prasad *et al.*, (2017) and the annotation of the collected tweets was done through crowd sourcing. Tweets were pulled composed with diversity to cover topics including race, gender, religion, educational background, and sexual orientation. Initially, 7,000 tweets were collected and annotated but on careful inspection, some of the tweets were found to be empty, duplicated and some gibberish necessitating preprocessing. The 500 Facebook comments collected were also subjected to the same cleaning and must of the comments were found worthless for the experiment and therefore deleted. Table 1 below shows the composition of the tweets/comments in the dataset.

Table 1: Polarity of comments/tweets in the dataset.

	Sarcastic	Non Sarcastic	Total
Tweets/Comments	237	5789	6026

First, we test the sarcasm detection framework in single domain, dividing the Twitter Sarcasm Corpus into training and test sets, with 80% training data and 20% test data. This step was carried out to ascertain the applicability of the proposed model to detect sarcastic comments/tweets. The training dataset was used to train ensemble method and SVM for the selected features. After training both the SVM and ensemble models using tweets they were then tested for sarcasm detection. These algorithms were also used for sarcasm type detection.

The following metrics were used to evaluate the performance of the model developed.

$$Precision = \frac{(True\ Positive)}{(True\ Positive + False\ Positive)} \tag{2}$$

$$Recall = \frac{(True\ Positive)}{(True\ Positive + False\ Negative)} \tag{3}$$

$$Accuracy = \frac{(True\ Positive + True\ Negative)}{(True\ Positive + False\ Positive + False\ Negative + True\ Negative)} \tag{4}$$

$$F - Measure = \frac{2 * (Recall * Precision)}{(Precision + Recall)} \tag{5}$$

The definitions of the terms used are as follows:

1. True Positive: Attributes predicted positive and its actually positive.
2. True Negatives: The attributes that are predicted negatively and are indeed negative.
3. False positives: False positives are wrongly predicted positive despite belonging to negative class.
4. False Negatives: False negatives are wrongly predicted negative while they belong to positive class.

For the sarcasm detection task, the results of sarcasm detection on the Twitter sarcasm corpus using SVM and ensemble method are as listed in Table 2. Both SVM as well as the ensemble classifiers performed well, with comparable F1 scores of 0.81 and 0.84 respectively, indicating their suitability for the sarcasm detection task. SVM reports higher Precision than the ensemble method but the ensemble method classifier gives more Recall. Overall, the ensemble method is better with higher Accuracy score of 0.94.

Table 2. Experimental Results of Ensemble and SVM Classifiers on Corpus of Tweets

Model	Precision	Recall	F1 Score	Accuracy
Ensemble Method	0.78	0.92	0.84	0.94
SVM	0.82	0.85	0.81	0.83

As we can see, combination of all the four feature categories reports the best performance. Lexical

features alone are not appropriate for sarcasm detection but when Pragmatic, Linguistic and Context Incongruity features are all added in the dataset they then become suitable for sarcasm detection and the combination of these feature categories with lexical features improve the performance of the sarcasm detection model as can be seen in the results.

In the case of sarcasm type detection, the results are listed in Table 3. Both SVM and the ensemble method give comparable F1 scores of 0.63 and 0.71 respectively. The result indicates that Recall values for both the classifiers are even better in the sarcasm type detection than in sarcasm detection task, indicating the proposed approach’s efficiency in detecting the type of sarcastic utterances out of the total sarcastic utterances in the dataset. In detecting the type of sarcasm, the ensemble model is having the highest F1 score signifying better accuracy.

These results establish that models trained on tweets can be used to detect sarcasm in texts including on product reviews. This is of importance in several NLP applications like sentiment analysis where there is a need to filter out sarcastic content and there may not be enough annotated sarcastic data for that domain and that level of granularity.

Table 3. Experimental Results of ensemble and SVM Classifiers on Sarcasm type

Model	Precision	Recall	F1 Score	Accuracy
Ensemble Method	0.52	0.96	0.71	0.85
SVM	0.55	0.87	0.63	0.79

4.0 Conclusions

In sentiment analysis, sarcastic expressions if not detected will affect the accuracy of the prediction model. In this study, we used SVM and an ensemble method with Random Forest, Naïve Bayes’ and k-Nearest Neighbour to conduct experiment to detect sarcasm and sarcasm type on social media text from Twitter and Facebook. The results obtained showed that in both tasks, the ensemble method has better accuracy compare to SVM. This show that Ensemble method is suitable for sarcasm detection task. The model can also be extended to product review, politics and stock market comments to diagnose the mood of the users.

Declarations

Ethics approval and consent to participate

Not Applicable

Consent for publication

All authors have read and consented to the submission of the manuscript.

Availability of data and material

Not Applicable.

Competing interests

All authors declare no competing interests.

Funding

There was no funding for the current report.

References

- Anukarsh, G. P., Sanjana, S., Skanda M. B., and Harish, B. S. (2017). "Sentiment Analysis for Sarcasm Detection on Streaming Short Text Data", 2nd International Conference on Knowledge Engineering and Applications, IEEE, 2017.
- Ashwin, R., Reza, Z., and Huan, L. (2015) "Sarcasm Detection on Twitter: A Behavioral Modeling Approach", *Proceedings of 8th ACM International Conference on Web Search and Data Mining*, P. 97-101, 2015.
- Bamman, D., and Smith, N. (2015). Contextualized Sarcasm Detection on Twitter Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org)
- Baruah, A., Arup, U., and Frank H. (2020). "Context-aware sarcasm detection using BERT." *Proceedings of the Second Workshop on Figurative Language Processing*. 2020.
- Bhakuni, M., Kumar K., Sonia, K., Iwendi, C., and Singh, A. "Evolution and Evaluation: Sarcasm Analysis for Twitter Data Using Sentiment Analysis," *J. Sensors*, vol. 2022, 2022, doi: 10.1155/2022/6287559.
- Du, X., James T., and Yu, W. (2022). "An effective sarcasm detection approach based on sentimental context and individual expression habits." *Cognitive Computation* 14.1 (2022): 78-90.
- Geng, Q., Feng, A., Zing I. (2022). "Multimodal sentiment analysis based on multi-head self-attention and convolutional block attention module." *2022 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*. IEEE, 2022.
- Kreuz, R. J., and Roberts, R. M. (1995). "Two Cues for Verbal Irony: Hyperbole and the Ironic Tone of Voice", *Metaphor and Symbol*, Vol. 10, No. 1, P. 21-31, 1995.
- Kumar, S., Raji, B., Avinash, U. (2020). "Sarcasm detection using multi-head attention based bidirectional LSTM." *IEEE Access* 8 (2020): 6388-6397.
- Liebrecht, F. K. (2013). The perfect solution for detecting sarcasm in tweets #not *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, P. 29–37, Atlanta, Georgia, 14 June 2013.
- Manoj, Y., Manohar, P. (2017). "Improvement Sarcasm Analysis using NLP and Corpus based Approach", *International Conference on Intelligence Computing and Control Systems (ICICCS)*, IEEE, 2017.
- Mondher, B., and Tomoaki, O. (2017). A pattern-based approach for sarcasm detection on twitter. *IEEE Transl.* 2017.
- Ortega-Bueno, R., Rangel, F., Hernandez-Farias, D., Rosso, P., Montes-y-Go´mez, M., Medina-Pagola, J.E. (2019). Overview of the task on irony detection in Spanish variants. In: *Proceedings of the Iberian languages evaluation forum (IberLEF 2019)*, co-located with the 34th conference of the Spanish Society for natural language processing (SEPLN 2019).
- Paras D., Tanupriya, C., Rajat, M., and Praveen K. (2017). Automatic sarcasm detection using feature selection. In *2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, P. 29-34. IEEE, 2017.
- Pornima, G. and Leena R. (2017) Sarcasm detection of non# tagged statements using mlp-bp. In *2017 International Conference on Advances in Computing, Communication and Control (ICAC3)*, P. 1-4. IEEE, 2017.
- Prasad, N., and Anukarsh G. (2017). "Sentiment analysis for sarcasm detection on streaming short text data." *Knowledge Engineering and Applications (ICKEA)*, 2017 2nd International Conference on. IEEE, 2017.
- Razali, M. D., Saifullah, G., and Hamis, J. (2020). "Sarcasm detection using deep learning with contextual features." *IEEE Access* 9 (2021): 68609-68618. Ren, Lu, *et al.* "Sarcasm detection with sentiment semantics enhanced multi-level memory network." *Neurocomputing* 401 (2020): 320-326.
- Santosh, K. B., Korra S. B., and Rahul R. (2017). Context-based sarcasm detection in hindi tweets. In *2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR)*, P 1-6. IEEE, 2017.
- Shrivastava, K., Mayank, S., and Shishir, K. (2021). "A pragmatic and intelligent model for sarcasm detection in a social media text." *Technology in Society* 64 (2021): 101489.
- Shubhadeep, M., and Pradip, K. B. (2017). Sarcasm detection in microblogs using naive bayes and fuzzy clustering. *Technology in Society*, (48) P. 19-27, 2017.
- Srivastava, K., Himani, L., and Hilad D. (2020). "A novel hierarchical BERT architecture for sarcasm detection." *Proceedings of the Second Workshop on Figurative Language Processing*. 2020.
- Wu, T., Yang, S., and Sing, J. (2021). "Modeling incongruity between modalities for multimodal sarcasm detection." *IEEE MultiMedia* 28.2 (2021): 86-95.