# A SURVEY ON SARCASM DETECTION APPROACHES

JIHAD ABOOBAKER
Research Scholar, Department of Computer science and Engineering,
Pondicherry Engineering College, Kalapet, Puducherry-605014, India
jihadabubacker@gmail.com

Dr E. ILAVARASAN
Professor, Department of Computer Science and Engineering
Pondicherry Engineering College, Kalapet, Puducherry-605014, India
eilavarasan@pec.edu

**Abstract - Natural Language Processing (NLP) is always one of the interesting topics among researchers. To understand the perfect meaning of what depicted in the conversation is always a helping factor to solve different tasks and enhance the accuracy of different applications. Sentiment analysis uses the NLP techniques and learning models like machine learning and deep learning algorithms to understand sentiments expressed in the given data. Sentiment analysis is an approach to find the contextual meaning expressed in the textual data. Sarcasm detection comes as part of sentiment analysis because sarcasm is a kind of sentiment where individuals convey their feelings about a particular topic indirectly. People means the entire opposite of the surface content of the sentence. This unique characteristic of the sarcastic sentence makes it difficult to plot sarcasm. This paper will discuss the works done in the area of sarcasm detection, different techniques and challenges in sarcasm detection.**

***Index Terms:*** Sentiment analysis; sarcasm detection; machine learning; deep learning.

## 1. Introduction

Sentiment analysis is the research area in which researchers' study and examine a person's sentiments, feelings, opinions, emotions, sentiment context, way of expressing sentiments about a topic or incident. The main studies carried out in the field of sentiment analysis are text-based or we can say these studies mainly focused on text mining. Sentiment analysis is one of the famous research areas in NLP (Natural Language processing). Sentiment analysis gained more importance in this era of social media. People use social media more frequently these days to express their feelings. Facebook, Twitter and Reddit are some of the famous social media platforms and these platforms became part of most people's life those who use the internet. Increase the usage of social media creates a vast quantity of unstructured data. We can say that the amount of this data increases exponentially in day by day. These data are always useful for various applications. Because people express their feelings about anything through these data. For example, if a person wants to buy something or want to see a movie, usually that person first search about it on the internet to know the opinions or reviews of other people about that particular product or movie. Many companies or even pollical parties want to know the current pulse of society. That's where the sentiment analysis and its importance come. Sentiment analysis can find or extract the feelings of the public. Sentiment analysis is a useful tool in different areas like politics, e-commerce, market intelligence and movie promotions etc., to know the public thoughts. For example, reviews given by customers who already brought a product in e-commerce sites like Amazon or Flipkart can be an influential factor for other customers to buy the same product. This importance of sentiment analysis also makes sarcasm detection an important matter. Sarcasm is a kind of sentiment where individuals convey their feelings about a particular topic indirectly. Sarcasm detection is known as the Achilles' heel of sentiment analysis. Because, if sarcastic utterance present in a sentence can change the actual meaning of the sentence and can change the result of sentiment analysis. The researchers Maynard and Greenwood [17] proved from their research that plotting sarcastic words effectively can enhance the performance of sentiment analysis. The word sarcasm originated from a Greek word, "*sarkasmos*". Cambridge Dictionary defines sarcasm as the use of remarks that clearly means the opposite of what they say, made in order to hurt someone's feelings or to criticize something in a humorous way. The Merriam-Webster dictionary defines sarcasm as a sharp and often satirical or ironic utterance designed to cut or give pain. It is easy to detect sarcasm from a verbal conversation than from textual data. When people use sarcasm in their conversation, the listener can understand the sarcastic words by noting the tone of the speaker, face expressions and body gestures. Lack of these clues in text sentences makes sarcasm detection from text data more difficult. Most of the people do not use formal language while they are posting something in social media sites. And also, they are keen to use local slang and abbreviations like "lol", means laughing out loud. This makes sarcasm detection from the text more complicated. So, to do efficient sarcasm detection, the proper training of learning models is an important fact. There are different approaches to detect sarcasm. The main categories in sarcasm detection approaches are Rule-

based, Machine-learning based and Deep-learning based approaches. Examples for Rule-based approaches are lexical, pragmatic and prosodic methods. In a lexical method, lexical features like adjectives, nouns and verbs etc., are considered. These features can extract by using n-gram approach. Punctuations play an important role in the pragmatic method. Because overmuch use of punctuations hints the presence of sarcasm. Prosodic features will appear when someone speaks in a connected speech. Stress, pause and rhythm etc., are the examples for prosodic features. Joseph Tepperman [6] used these prosodic features for sarcasm detection. For machine learning-based approaches, different classifiers like SVM (Support Vector Machine), Random Forest, Naïve Bayes and Decision tree etc., can be used. CNN (Convolutional Neural Network), RNN (Recurrent Neural Network) and LSTM (Long Short-Term Memory) etc., are using for deep learning-based approaches for sarcasm detection. In recent years, deep learning and machine learning approaches are mainly used for sarcasm detection tasks. Various datasets are using for sarcasm detection, among Twitter dataset plays an important role. The main purpose of this survey paper is to give details about past works in sarcasm detection to help researchers to understand the work done in this area. This paper divided into different sections like literature review, general architecture and challenges etc.

## 2. Characteristics of Sarcasm

John D. Campbell *et al* [43] studied the different contextual components used to deliver the sense of sarcasm. They stated that the sarcasm takes place along different dimensions like the presence of a sufferer, failed expectation and negative tension etc. Deirdre Wilson [47] attempted to explain verbal irony. They suggested that sarcasm occurs whenever a circumstantial disparity arises between the contextual meaning and the text. For example, consider the statement "I love being neglected " is considered as a sarcastic sentence because of the great difference between text and contextual meaning. The contextual information from the sentence is 'being neglected' is considered as an unpleasant situation. But the speaker of the sentence stated that to love this unpleasant situation. Jodi Eisterhold *et al* [36] suggested that the sarcasm can be identified by reaction or response from the listeners. From their study, they suggested that certain body gestures, laugh, changing of the topic etc., are the reactions which indicate the presence of sarcasm. Elisabeth Camp [54] four different classes of sarcasm. They are:

### *2.1. Illocutionary sarcasm*

Illocutionary sarcasm is a type of utterance as the form of warning od suggestion etc. In this type of sarcasm, non-textual clues will be present and which shows an attitude contradicts to sincere utterance. Consider the statement "Thanks for holding the door". The speaker said this dialogue to a person who just shut a door forcefully and loudly towards the speaker.

### *2.2. Like-prefixed sarcasm*

This type of sarcasm uses a like-phrase. This like-phrase gives an indirect refusal of the utterance being made. For example, "Like that's a good idea!".

### *2.3. Propositional sarcasm*

This type of sarcasm contains a proposition. Proposition means a declaration that shows an opinion. This sarcasm is a more straightforward type of sarcasm due to the presence of proposition. The understanding of context is important in the case of propositional sarcasm. Propositional sarcasm contains indirect sentiment. For example, "He is a fine friend!". Without knowing the context of the sentence, this sentence may be considered as a non-sarcastic sentence.

### *2.4. Lexical sarcasm*

Lexical sarcasm provides reverse compositional importance to a single phrase or word. Consider the sentence, "Because John has become an honest diplomat, nobody considers him". The word 'diplomat' in the sentence has inverted the whole meaning of the sentence. Stacey L. Ivanko and Penny M. Pexman introduced a six-tuple representation for sarcasm. Their suggested approach consisted of the following six tuples <S, H, C, u, p, p'>. S is the 'Speaker', H is 'Hearer', or 'Listener', C is 'Context', u is 'Utterance', p is 'Literal proposition' and p' is 'Intended proposition'. The above six-tuple representation can be interpreted as: The Speaker (S) produces an Utterance (u) in the Context (C) which means the Literal proposition (p). But actually, expected that the Hearer (H) should interpret the actual Intended proposition (p'). Consider the sentence, "I love being neglected". We can represent this sentence in six tuple representation as,

S = The speaker referred as by 'I'

H = The hearer

C = Context

u = "I love being neglected"

p = "I love being neglected"

p' = "I hate being neglected"

## 3. Literature review

Yufeng dieo *et al* [10] proposed a novel multi-dimension question answering (MQA) network to find sarcasm. The proposed architecture is based on Bi-LSTM (Bi-directional long short-term memory) and attention mechanism. Bi-LSTM is an RNN (Recurrent neural network) architecture. RNN is one of the famous deep learning models using for applications like text summarization, predictions problems etc. The attention mechanism is one of the good concepts in the deep learning area of work. Attention mechanism works on the principle that giving attention to some relevant topics and ignoring other topics which are not relevant based on the goal. The researchers proposed a question answering framework based on Bi-LSTM and attention mechanism to improve the detection of sarcasm. Word embedding is a technique for feature extraction in which words are converted into a vector representation. So, the words with similar meanings have a similar vector representation. Word2Vec and Glove are the famous word embedding techniques. In this work, researchers used Glove mechanism for word embedding process. They choose LSTM due to its speciality to create the sequential property of text data. They used IAC V2 (Internet argument corpus) data set. IAC V2 is a famous data set which consists of the online social media forum conversation. They opted f-score, recall and precision as their evaluation metrics. They compared their proposed framework results with some machine learning methods like support vector machine, random forest classifier and logistic regression classifier. Also, they compared their results with basic LSTM model. They concluded that their proposed MQA model outperforms above-mentioned techniques based on the evaluation metrics results.

Himani and Vaibhav *et al* [46] proposed a novel deep learning-based method to detect a sentence is sarcastic or not based on the circumstances of that particular sentence. They used Reddit and Twitter data set for their proposed work. The researchers presented a hierarchical BERT based model for detecting sarcasm. BERT (Bidirectional Encoder Representations from Transformers) is a neural network technique which is widely using for NLP (Natural language processing). The main advantage of BERT is that it makes a language model understand the context or circumstances of a word based on its surrounding words. The researchers first tried to extract the local features of words in a sentence. When the extraction of features is done, they used CNN (Convolutional neural network) layer to obtain the relationship between response and its context. From the features they extracted, they used a Max-pool layer to extract the most important features. To find the probability score to determine a sentence is sarcastic or not, they used the sigmoid layer. Their two data set contains the conversation between the users, which acted as the context. They tried to find the final response in conversation is sarcastic or not based on the context. For evaluation purpose, they used the F1 score. They concluded from their work that; the results can be better in the context-based approach. Edoardo *et al* [45] proposed work to detect sarcasm from discussion forums and social networking platforms using the possibilities of multi-tasking learning and deep neural networks. In their proposed multi-task learning framework, sentiment analysis comes as an auxiliary task to support the sarcasm detection. They used Bi-LSTM withs sentiment classification to enhance sarcasm detection. They used the SARC data set which consists of sarcastic comments from the Reddit site. Reddit is a social networking platform in which users can discuss different topics. They used Multi-Layer Perceptron (MLP) in their architecture. MLP is an Artificial Neural Network (ANN) class which consists of feed-forward ANN. For sentiment analysis purpose they used pre-trained architecture for analysis of sentiments which is available on Stanford NLP site. They assigned one of the four labels to each sentence. Those four labels are very-positive, negative, very-negative and positive. From their work, they concluded that when compared to multi-tasking Bi-LSTM can obtain better results than normal Bi-LSTM model. They used F1-Score as their evaluation metric. Shawni *et al* [12] proposed a novel framework for detecting not only sarcasm but also the sarcastic users. The researchers did a study on various sarcasm detecting techniques in text data. They meant by sarcastic users as the users those who are more likely to use sarcasm. Their proposed architecture consists of two steps. In the first step, they are identifying tweets are sarcastic or not. Based on the first step, they are doing identification on sarcastic users as in the second step. They are using Twitter API for collecting tweets. To form feature vectors, they extracted lexical, semantic and syntactic feature. These feature set give as the input to RNN (Recurrent neural network) for the analysis content of tweets. After getting the output from RNN, they again give those output to another RNN network to give some score to each tweet. Based on these scores they did label the tweet as sarcastic or not. After this step, to recognize a user as a sarcastic user or not process begin. For that, they used the tweets belonging to each user and then repeat the works in the first step. After based on the tweet score, they compared this score with a predefined value and label users as sarcastic users or non-sarcastic user. Adithya *et al* [3] proposed that the BERT is giving better performance to extract semantic features from conversations compared to other models. They performed a comparative study with different types of LSTM models and models based on BERT. From their study, they showed that the BERT based models outperformed other models. They used F-measure as their evaluation metric and they achieved 0.752 F-measure on Twitter data set and 0.621 F-measure for Reddit data set. Their Twitter and Reddit data sets consisted of three different fields. They are; context, label and response fields. They did experiments on LSTM, CNN-LSTM, Bi-LSTM and stacked LSTM. Underfitting and overfitting are the problems faced by researchers when they are training the models with training data sets. Overfitting happens when a model understands more about noise in the data sets which produces a negative effect on the working of that model. To

avoid overfitting, the researchers of this work adopted the dropout layer mechanism. In dropout layer mechanism randomly drops some layer outputs of neural networks.

Sayed and Agarwal *et al* [44] proposed a work based on deep LSTM-RNN with word embedding techniques for detecting sarcasm from sentences. They applied the word embedding technique on pre-processed data sets. They used Twitter API to form their dataset. They mainly used tokenization and sequence padding techniques for their pre-processing works. Tokenization means to convert given sentences or text into words. Different sentences in the dataset have different lengths. For the better performance of the classification model, the input should be of the same length. Sequence padding is a technique to achieve this goal. So, the researchers of this paper used the sequence padding technique to make each tweet in the dataset to have equal length. To overcome the overfitting problem, they implemented the dropout layer method. From their work, they reached on the conclusion that RNN-LSTM model is better than SVM or any other machine learning algorithm when it comes to the detection of sarcasm from tweets. Arup and Kaushik *et al* [4] done work based on the SVM, BERT and Bi-LSTM model for detecting sarcasm and compared the results from each classifier. They concluded their analysis as, whenever the use of the last utterance of the sentence with the response to that sentence can boost the performance of the classifier to detect sarcasm from the Twitter dataset. They also found that in the case of Reddit dataset, analysing only the response sentence can improve the results. They chose one Twitter data set and one Reddit dataset. For Bi-LSTM model, they used FastText word embedding technique. FastText is a word embedding technique introduced by the Facebook research team. For SVM, they used TF-IDF (Term Frequency-Inverse Document frequency) for feature extraction. They used 5 cross-validations on models to validate the output. Cross-validation is an approach to check the effectiveness of the models. From their work, they found that the BERT model reached the highest F-score of 0.743 for the Twitter dataset when the last utterance and its response from a conversation are present. Also, BERT gives highest F-score of 0.658 on Reddit dataset when only the responses from the conversation are included to train the model. Neha *et al* [41] aimed to propose a model to detect automatically the sarcastic tweets from the given Twitter dataset. They used K-Nearest Neighbour (KNN), Random Forest and SVM classifiers for their work. Their dataset consisted of 9104 tweets with #sarcasm and #not hashtags. They removed URLs, white spaces etc., from their dataset. They mainly extracted the features related to sentiment, patterns, semantics, syntactic and punctuations. For cross-validation purpose, they applied the 10 cross-validation technique. From their work, they found that the random forest classifier achieved the highest accuracy, precision and F-score among the three classifiers. And SVM attained the highest recall value among the three classifiers. Amit *et al* [21] proposed a model named Contextual network (C-Net). The C-Net sequentially uses the contextual information from a sentence and uses this information to classify the sentence as sarcastic or not. They did their work as the part of the second workshop on figurative language processing of the Association for computational linguistics (ACL) 2020. They used one Twitter dataset and one Reddit dataset for their proposed work. They used SVM, Logistic regression, Naïve Bayes, RNN, Bi-LSTM, BERT and RoBERTa models for their work. For word embedding process, they used Glove method. Their two datasets have an equal number of non-sarcastic and sarcastic responses. In their proposed model C-Net, they used pseudo-labelling approach. In pseudo-labelling approach, context sentences have given the same labels as the particular response for each sentence. They opted approach to give importance to the contextual background to enhance the sarcasm detection from sentences. Then they trained the models. They divided their training dataset to 90% as for training and the remaining 10% for validation. For the pre-processing task, they used spacy tokenizer and torchtext library. From the results of their work, they reached on the conclusion that the C-Net model gave the F-score of 0.750 on the Twitter dataset and 0.663 on Reddit dataset when C-Net associated with SES (Simple Exponential Smoothing). Taha *et al* [20] proposed an approach based on aspect-based sentiment analysis and BERT to find the relationship between dialogue sequence and its response. Based on this they classified the response as non-sarcastic and sarcastic. They used Twitter dataset and Reddit dataset. They used the utterance and the history of conversation as input. Their Twitter data consisted of 6800 data samples and Reddit dataset consisted of 6200 dataset samples. Their dataset has four columns as follows. ID, Label, Response and Context. They used Glove and FastText methods for word embedding purpose. They trained their dataset on the following models. NBSVM (Naïve Bayes-Support Vector Machine) which is a combination model of Support Vector Machine and Naïve Bayes, Combination of BERT and SVM (BERT-SVM), XLNET, a combination of BERT and Logistic regression (BERT-LR) and LCF-BERT (It is an aspect-based model for the classification of sentiments) etc. They concluded from the result of their work that; LCF-BERT model is having the highest F-score (0.730) among the models when Reddit dataset is used to train the models. Their results showed that BERT model and the combination of BERT and aspect-based sentiment analysis approach gives better results in sarcasm detection on datasets.

Ameeta *et al* [1] proposed a method in which they analysed the transition on change in various emotion over the course of a piece of text. They named their model as EmoTrans (Emotion Transition). They first divided the sentences into non-overlapping chunks. Then for these chunks, they formed a vector of emotion features. This emotion vectors then gave as the input to sequence classification models. Their main goal was to find the transitions of emotions within the text and find the effective features from the chunks of text and train the model with these features to find sarcasm effectively. For the chunking of text, they used phrase-based chunking, Fixed-n chunking, Fixed-k chunking approaches. To obtain the emotion vector of a chunk, they used WNA (WordNet Affect), WEES (Word-Embedding based emotion scores) and NRC (NRC EmoLex) methods. They used the LSTM model as the classifier. They used two sarcasm datasets. Onion news headlines dataset and IAC debate corpus dataset. From their work and comparing other baseline models with their proposed EmoTrans, they found that their proposed approach outperformed some of the baseline models. Vaishvi *et al* [19] proposed work to obtain optimal features before the data is passed to classifiers. By doing this, the goal of the researchers was to boost the performance of the classifier. They used an SVM classifier model to detect sarcasm from news headlines. Their dataset consisted of sarcastic and regular news headlines. The sarcastic news headlines collected from the Onion news network and the normal news headlines collected from Huffpost news network. The dataset consisted of 27000 headlines. In which 11700 are sarcastic and 14900 are non-sarcastic. As the part of data pre-processing, researchers did tokenization, stop word removal, stemming and lemmatization techniques on the dataset. As part of feature extraction, the proposed system extracted a total of 17 features. Some of them are unigram, positive and negative intensifiers, verb count etc. SVM was chosen as the classifier for the proposed work to its effectiveness of binary classification. The dataset was collected from Kaggle site. As evaluation metrics; precision, recall and F-score are considered. The research concluded from their work that the SVM improves its performance when the optimal feature set was given as input.

Avinash and Vishnu *et al* [28] introduced an MHA-BiLSTM (Multi-Head Attention-based Bidirectional Long-Short memory) network to detect sarcasm efficiently from a given dataset. Also, they built an SVM model for sarcasm detection. They considered different hand-crafted features. Both their models used these handcrafted features to training purpose. The significant of this MHA-BiLSTM model is to identify the significant factors of a sentence which can help to increase the detection of sarcasm. They reproduced different state-of-the-art model on their datasets to compare the results of the above-mentioned model and their proposed model. They used SARC to form their datasets. SARC mainly consisted of sarcastic and non-sarcastic comments from the Reddit site. They used precision, F-score and recall as their evaluation metrics. As for the pre-processing part, they removed stop words, URLs etc. They randomly divided their 90% training set for training and 10% for validation purpose. Their proposed work MHA-BiLSTM has two main layers. One is word encoder layer and another one is sentence-level multi-head attention layer. The word encoder layer summarizes the contextual information of each word in comment gives each word a new representation. On the other hand, sentence-level multi-head attention layer, at the same time gives the attention to the different parts of comments to notice the different semantic aspect of the comment. From their work, they found that the manually created auxiliary features enhance the performance of the model. Also, they found that using the auxiliary features, MHA-BiLSTM gives better performance compared to other models. Jens *et al* [31] presented an ensemble-based approach to finding the sarcasm from Twitter and Reddit datasets. They presented their work as the part of the second workshop on figurative language processing held in conjunction with ACL 2020. In the ensemble learning approach, different classifiers are systematically generated and combined to find the solution for a computational problem. In this work, researchers used four component models. Also, they used additional features like the sentiment of a comment, source of the sentence etc., to find the most accurate component model for a given input. The four component models are SVM model, LSTM model, CNN-LSTM model and MLP (Multi-layer perceptron) model. Their Twitter dataset consisted of 5000 tweets and Reddit dataset consisted of 4,400 comments. Apart from the SVM model, other component models used the context of the conversation as a feature. SVM concentrated on emotion-based characteristics and stylometric properties. Stylometry is an approach which finds a difference in literary style between writers. For the training purpose of the component models, they used 10-fold cross-validation technique. From the results, they concluded that the ensemble model has higher precision, F-score and recall value.

Nikhil Jaiswal *et al* [18] proposed a deep neural architecture for sarcasm detection. They analysed different PLRMs (pre-trained language representation models) like RoBERTa, BERT etc. They used Twitter dataset. They tried to use the contextual information of the twitter utterance. They found that considering the previous three most recent utterance can enhance the ability of a model to clarify the conversation into sarcastic or not. They used the ensemble-based model approach. They did their work as part of the shared task on sarcasm detection 2020. They considered F-score, precision and recall as their evaluation metrics. They used the majority voting ensembles technique to find a good performance model. Different models they used are USE (Universal Sentence Encoder), ELMo (Embeddings from language model), BERT and RoBERTa (Robustly optimized BERT approach). Their dataset consisted of 5000 English tweets for training purpose and 1800 English tweets for testing purpose. From the experiments they did on the above-mentioned models, they concluded that providing more context history along with the utterance, the model can classify more effectively. Also, they found that the

RoBERTa model is giving better performance when comparing F-score, precision and recall compared to other models. Siti *et al* [26] proposed work to detect sarcasm from tweets by considering the varying context of tweets related to the word or phase in it. They used paragraph2Vec approach for word embedding purpose. Using this paragraph2Vec, they found the contextual meaning. The features extracted from using paragraph2Vec, they trained LSTM for classification purpose. They used twitter dataset of English and Indonesian languages. They collected both datasets using Twitter API. They used F-score, accuracy, precision and recall as their evaluation metrics. From the result, they concluded that by using context features with the help of paragraph2Vec and LSTM for classification, the accuracy of Indonesian dataset is 88.3% and 79% English dataset.

Akshay *et al* [24] analysed the different machine learning techniques with the Glove and BERT word embedding methods to find the sarcasm in tweets. Their proposed model also used contextual information. Glove embedding and BERT embedding are word embedding techniques to convert words to vector representations and for obtaining features from datasets. They used different pre-processing techniques like tokenization, stop word removal and stemming. They used classifiers like LSVC (Linear Support Vector Machine), Random forest, Gaussian Naïve Bayes and Logistic regression for their work. They used Scikit-learn to train these models. Scikit-learn is a machine learning library for Python. They used F-score as evaluation metrics. From their work result they concluded that, with the BERT word embedding technique, Logistic regression gives the best f-score compared to other models. And with Glove word embedding technique, logistic regression able to gives the best F-score compared to other models. Also, they concluded that considering context information for training models can boost the results. Kartikey *et al* [40] did work on Reddit and Twitter datasets. They used RoBERTa model to find sarcasm from both datasets. To improve the importance of contextual information, they used three types of inputs. They are; Context- Response input, Response-only input and Context-Response(separately). They suggested that adding a separator token between context and response increased the F-score in Reddit dataset. Their main goal was to use the possibility of contextual word embeddings for detecting sarcasm in Twitter and Reddit dataset. As mentioned earlier, they used three types of input to learn the effect of context on the performance of RoBERTa to detect sarcasm. They used F-score, precision and recall as their evaluation metrics. From their work, they reached on the conclusion that the addition of contextual information to target response and adding a separating token between context and target response can improve the performance of RoBERTa model. Amardeep *et al* [27] worked on latest pre-trained transformers like BERT, spanBERT and RoBERTa. RoBERTa is a model based on BERT. RoBERTa modified the hyperparameters in BERT and also it can train with the higher amount of data compared to BERT. SpanBERT model is also based on BERT model. The researchers of this paper also presented their model which includes LSTM and Transformers. They used Twitter and Reddit datasets. They divided their classification task into two different categories. Sentence-pair classification task and single sentence classification task. To get a single sentence for classification, they used the combination of context string and the response string. For the sentence pair classification task, they fed a pair of text as the input for classification. They introduced their Siamese Transformer model and Dual Transformer model. They concluded their work that the context information is necessary for the better performance of sarcasm detection model. Rahul and Harsh *et al* [16] divided their work into two phases. In the first phase, they worked on to extract features related to extraction and punctuation, and then applied the chi-square test to find the most important features. In the second phase, they extracted 200 top TF-IDF (Term Frequency- Inverse document frequency) features are extracted and then these features are combined with punctuation related features and sentiment related features to detect the sarcasm from the tweets. They used famous libraries like Pandas, NumPy, NLTK etc., for their work. Their dataset consisted of 16000 tweets. Chi-square test is used to extract the most important features from extracted features. They trained different machine learning models like KNN, SVM, Decision tree and Random forest with above mentioned most important features they selected based on chi-square test. They compared the accuracy of above-mentioned machine learning models. For the second phase, the combining of features was done. In the second phase, they used voting classifier model. It is a machine learning model which trains on a group of (ensemble) different models and find the output based on the highest majority voting. From their work, they concluded that the selection of only important features can improve the accuracy of models. From the result of the second phase of their work shows that the voting classifier (ensemble model) gives the higher performance.

Xiangjue *et al* [11] presented a transformer-based model for detecting sarcasm from the Twitter dataset and Reddit dataset. They used context-related information from the entire conversation to obtain better results to predict the sarcasm. Their model used deep transformer layer to carry-out multi-head attention amongst the important context and the target utterance. Their model used a transformer encoder to create the embedding form of the target utterance and its context by applying multi-head attentions. From their work, they concluded that the model shows better improvement in detecting sarcasm from given datasets only when considers the relevant context. They used two types of transformer-based sarcasm detection models. One model which takes target utterance as input. This model is target-oriented. Other models take the context of utterance as well as the target utterance as input. This model is the context-aware model. They used BERT-large, ALBERT-large and RoBERTa-large models. Above mentioned models are BERT based models. They trained all the three models on a combined Twitter and Reddit

dataset. From their work, they found that the context-aware model formed using the RoBERTa-large depicts better performance than other models.

Kalaivani *et al* [48] did their work to understand how important is conversion context or response for detecting sarcasm from Twitter and Reddit conversation datasets. They trained different machine learning and deep learning methods to find sarcasm. They used different libraries like NLTK and Gensim for data pre-processing. Gensim is a library using for natural language processing. They utilized the word cloud method to find the most important sarcastic words. Word cloud or text cloud is a method to find the more important or specific word which presented in the textual dataset. Word cloud represents a word in different sizes. More important words represented in bigger and bolder letters. For feature extraction, they used TF-IDF and Doc2Vec methods. Doc2Vec is a tool or model used to represent the collection of words in vector form. For the baseline machine models, they have chosen Gaussian Naïve Bayes, RandomForest and Logistic regression etc. As for deep learning-based models, they used RNN-LSTM model and for word embedding, they have chosen the BERT model only. From their result, they found that the BERT performs better than baseline machine learning models and RNN-LSTM model. Darkunde *et al* [2] proposed a model which is based on the optimization of parameters in LSTM by using genetic algorithm. Their model consisted of different layers like Input layer, Embedding layer, LSTM-Genetic optimization layer and CNN layer. Embedding layer converts the textual input data from the input layer to vector form. BERT model is used for this embedding purpose. For optimizing parameters for LSTM, Genetic optimization algorithms are used. Their proposed model mainly depends on lexical features. Their proposed model achieved an accuracy of around 93-95%. Chia Zheng *et al* [32] compared different pre-processing methods effect on the suggested deep convolutional neural network model. Their main goal was to analyse the difference in sarcasm detection results when different pre-processing techniques which are typically using for data pre-processing in NLP. They used Twitter dataset for their work. They classified pre-processing methods to different categories like, with hashtags, without hashtags, stemming applied, stop words removed, stemming and stop words removed, POS tagging applied and Hashtags, URLs, tagged users are removed. From their work, they reached on the conclusion that the better results of sarcasm detection achieved only when hashtags are present in the dataset. Le Hoan Son *et al* [29] proposed sAtt-BiLSTM convNet deep learning model. sAtt-BiLSTM stands for soft-attention based bidirectional long short-term memory and convNet stands for the convolutional neural network. They used Glove technique for word embedding. They used the balanced and unbalanced dataset to evaluate their proposed model. They used SemEval 2011 Task 11 dataset as balanced dataset and around 40000 random tweets are used as an unbalanced dataset, in which 15000 sarcastic tweets and 25000 non-sarcastic tweets. They compared their proposed model results with the convolutional neural network, LSTM and BiLSTM model results. Their proposed sAtt-BiLSTM convNet model consisted of different layers like Input layer, Embedding layer, BiLSTM layer, Attention layer, Convolutional layer, Activation layer, Downsampling layer, and Representation layer. They produced Vector matrix by using Glove and fed this matrix as the input to BiLSTM layer. BiLSTM layer can learn high-level features from the input vector matrix. In the convolution layer, the convolution operation is taking place to get a convolved feature vector. Activation layer contains ReLU (Rectified Linear Unit) activation function. ReLU is one of the famous activation functions used in deep learning methods. Activation functions are using to determine whether a neuron should be activated or not based on the output of the neuron given to activation functions. For downsampling technique, researchers of this paper used the max-pooling method. Max-pooling helps to solve the over-fitting problem. Representation layer is to produce output predictions based on the softmax activation function. They used Accuracy, Recall, Precision and F-score as their evaluation metrics. They concluded from their work that their proposed model outperformed other models by giving an accuracy of 91.60 % for SemEval dataset and 88.28% for random tweets.

Nan XU *et al* [52] proposed an approach for multi-modal sarcasm detection. They named their model as D&R Net (Decomposition and relation network). The decomposition network is for finding the commonality and differences between text and image in tweets. The relation network is for finding the semantic relationship between text and image. To find these relationships, researchers used a cross-modality attention mechanism. They compared their model with the state-of-the-art models in the multi-modal sarcasm detection. Their model consisted of four modules. They are pre-processing, encoding, relation network, and decomposition network. They first extracted ANPs (Adjective-Noun pairs) from each image. Then they found the differences and commonality between the text and images. They used Glove model for word embedding. To extract features from images, they used pre-trained ResNet. To extract relevant ANPs, they used SentiBank toolkit. They used F-score as their evaluation metric. Their model showed higher F-score compared to other models like ResNet and CNN. Nirmala *et al* [38] proposed a work to create an unsupervised probabilistic relational framework to recognize common sarcasm topics. They analysed the distribution of sentiment of the words in tweets to find common sarcasm topics. The reason to find common topics because some topics within the tweets are more biased to be sarcastic when compared to other topics. They used Twitter API to download tweets. They used F-score, Accuracy, Precision and recall as their evaluation metrics. They collected 150000 sarcastic tweets and non-sarcastic tweets of size 300000. As the part of pre-processing, they implemented stop word removal, converting characters of words into lowercase etc. They analysed Incongruity, Pragmatic, Lexical and subjective features. They named their model as

(Sentiment Topic Sarcasm Mixture Model). Their proposed model showed higher precision and recall results and a better F-score measure.

Hongliang and Zhong *et al* [39] proposed the SAWs (Self-Attention of Weighted Snippets). They mainly tried to solve the snippet incongruity problem in sarcasm detection. A snippet is a small portion of a text. Incongruity or discordance means something is not right or something is not fitting right. So, finding this incongruity or mismatch between the meaning of the sentence and the actual meaning hints the presence of sarcasm. But this mismatch may not be always present in every sarcastic sentence. That is the reason, the researchers tried to consider the mismatch or incongruity between the sentence snippets. They used Tweet datasets and IAC (Internet Argument Corpus) dataset. Their model consisted of an input module, importance weighing module, a convolution module and a self-attention module. They introduced a context vector to find the importance of snippets from a given sentence. This is because not all the snippets are that much important to detect the sarcasm. The important snippets will have high weights. They used Glove for word embedding. From their work, they concluded that their SWAS model improved the F-score of sarcasm detection process in a given set of datasets compared to other models like CNN-LSTM-DNN model. Hankyol Lee *et al* [30] proposed contextual Response Augmentation (CRA) model. CRA creates meaningful samples for training purpose by the help of the conversational context of the text. They also addressed the issues of unbalanced context length of the data. They change the format of input-output. So that models can effectively handle the variation in context lengths. They proposed a context ensemble method to train the model. The main components in their model are BERT, BiLSTM, NetXtVLAD etc. NetXtVLAD is a CNN-based model which can effectively handle the over-fitting problem. They used Recall, Precision and F-score as their evaluation metrics. From their work results, they understood that the F-score of sarcasm detection increased when data augmentation is applied. They implemented labelled and unlabelled augmentation techniques. Data augmentation is a technique that increases the variety of dataset for training purpose without gathering new data.

Jacob *et al* [9] introduced Bidirectional encoder representations from Transformers (BERT). BERT is a language representation model. The main goal of BERT is to pre-train bidirectional representations from the unlabelled text by simultaneously considering both right and left context of the input data. BERT can be used for a variety of tasks in NLP like language understandings, analysing question answering etc. the researchers of this paper trained their BERT model on the different NLP tasks like GLUE and SQuAD v1.1 question answering etc., to check the performance of BERT. The researchers pointed out that the major drawback of normal language analysis models is that they are unidirectional. Means they are left-to-right architecture models. For example, in a left-to-right model, every word taken is only giving attention to the previous word tokens. To overcome this, researchers used an MLM (Masked language model) approach in BERT. The MLM randomly masks a few tokens from the input data. Then tries to find or predict the vocabulary id of this masked token word based on its context. This approach helps BERT to pre-train a deep bidirectional transformer. Together with MLM, researchers used another method called next sentence prediction which pre-trains the text-pair representations. The researchers of this paper claimed that the BERT is the first model based on fine-tuning that attained state-of-the-art performance on large token-level tasks and sentence level tasks which outperformed several task-specific architectures. There are two main steps in BERT. One is pre-training and the other one is fine-tuning. In the pre-training phase, BERT is trained over various pre-training tasks with unlabelled data. In the fine-tuning phase, BERT is prepared with pre-trained parameters. The pre-trained parameters are then fine-tuned with labelled data from the downstream tasks. Downstream tasks are the supervised learning task that uses a pre-trained architecture or component. To prepare the BERT model to handle the different downstream tasks, the input representation given to BERT should be able to represent both a pair of sentences and a single sentence. The researchers used WordPiece embedding method. They used BooksCorpus and English Wikipedia corpus as pre-training corpus. BERT used self-attention mechanism for fine-tuning. Fine-tuning of BERT is done by using different NLP tasks like GLUE (General Language Understanding Evaluation), SQuAD v1.1 (The Stanford Question Answering Dataset) task, SWAG (Situation With Adversarial Generations) dataset etc. The researchers concluded from their work results that their proposed BERT model outperformed several baseline models. Also, they claimed that the BERT model performed better with different NLP tasks when compared to some other existing models.

Zhilin *et al* [53] proposed XLNet architecture which is an autoregressive pretraining model. XLNet is for understanding bidirectional contexts by enhancing the anticipated probabilities over all permutations of the factorization order. It also overcomes the limitations of BERT by using autoregressive formulation. The researchers claimed that the XLNet outperformed the BERT model on 20 different NLP tasks. AR (Autoregressive) language modelling and AE (Auto Encoding) language modelling are the two pretraining approaches. Purpose of AR is to find the probability distribution of the given text corpus with the autoregressive model. AE works on the restoration of the original data from the changed input. XLNet uses the advantages of the AR and AE language models to enhance its performance. XLN*et al*so extracts the bidirectional contexts of the words. XLNet keeps the original sequence order of the data and uses an appropriate mask in transformers to get the permutation of the factorization order. XLNet learns the dependency pairs from the given data. XLNet used

BooksCorpus and English Wikipedia for their data pre-training. Also, researchers of this paper included the ClueWeb 2012-B, Common Crawl and Giga5 text corpora for data pre-training for the model. Main components of the XLNet architecture are Content stream attention layer, Query stream attention layer, masked two-stream attention layer. The XLNet model trained on different NLP tasks like GLUE, SQuAD, RACE and IMDB etc., and obtained better performance results. The researchers concluded that their proposed model outperforms some existed models. Rishab *et al* [37] introduced a new dataset which consisted of sarcastic news headlines and non-sarcastic news headlines. They proposed a hybrid neural network model with an attention-based mechanism. The main goal of this work is to find the factors which makes a sentence to a sarcastic sentence. They suggested that to detect sarcasm effectively, understanding of common-sense knowledge is important for models. To create the proposed dataset, they collected the news headlines from The Onion News website and HuffPost news website. They used the word cloud method to find the frequently occurring words in the dataset. Their proposed architecture consisted of BiLSTM module, Attention module, CNN module and MLP module. They used word2vec for word embedding. They concluded that their proposed model outperforms the baseline models in terms of accuracy.

Tao *et al* [51] proposed a novel self-matching network architecture to analyse the word to word connection and to find the sentence incongruity. Their proposed model analysed each word-to-word pair from the input sequence and calculated joint information for each pair to create self-matching attention. Based on this attention vector, altered the sentence and created its representation vector. They also included BiLSTM module in their proposed network to get the compositional information. They linked together with the compositional information and the incongruity information by using a Low-rank Bilinear Pooling method to restrict the redundancy in potential and at the same time not dropping the discriminative power. They used co-attention mechanism in their proposed self-matching network to enhance the word-to-word comparison. This method helped their model to capture incongruity information occurred due to sentiment conflicts between words. They evaluated their model using different publicly available datasets. They concluded their experiment result that their proposed self-matching network model outperformed some sarcasm detection models based on the neural network when comparing models with standard evaluation metrics like Precision, Recall, Accuracy and F-score. Other findings from their work are that employing a sequential network like LSTM can help the self-matching network to improve its performance. They used Reddit, IAC and Twitter datasets for their experiments. They used CNN-LSTM-DNN and ATT-LSTM models etc as their baseline models. Yinhan *et al* [34] proposed a replication study on BERT model and proposed their model based on BERT named as RoBERTa (Robustly Optimized BERT). They studied BERT model and analysed the impact of different parameters and training dataset size on the model. They claimed in this paper that their model achieved state-of-the-art results on the NLP tasks like GLUE, SQuAD, RACE. They found from their study that the BERT model is undertrained. They improved the BERT model and proposed RoBERTa model. Their modification on BERT model includes the following main changes. They trained the BERT model with longer datasets with bigger batches. They removed the objective of next sentence prediction. And also, they trained the model with a longer sequence and changed the masking pattern dynamically and then applied it to the data. They introduced CC-NEWS dataset which is a large new dataset. They claimed that their model established a new state-of-the-art approach for different tasks like GLUE, RACE and SQUAD. They suggested that by selecting correct design choices, the masked language pre-training approach is very effective. They used BooksCorpus, OPENWEBTEXT, CC-NEWS etc. They evaluated RoBERTa model using SQuAD, GLUE, and RACE (the Reading Comprehension from Examinations) benchmark datasets.

Lu Ren *et al* [42] used sentiment semantic to obtain the difference in sarcasm expression features and proposed a multi-level memory network. They used the first level memory network to plot sentiment semantic. To capture the difference between the situation in a sentence and the sentiment semantics, the researchers used the second level memory network. Sometimes the local information may be absent in the sentence. In the case of the absence of local information, they used a enhance model of CNN. IAC (Internet Argument Corpus) and Twitter dataset for their work. They named their model as MMNSS (Multi-level Memory Network based on sentiment semantics). They used local max-pooling layer in the CNN model rather than the traditional max-pooling layer which can successfully maintain useful features. They mainly considered three types of information from the sentences. They are contextual information, local information, and the information which can point out the difference of sentiments in the sentence. Their model consisted of different components like first level memory network, second level memory network, CNN module, LSTM module and MLP module. They used Glove as their word embedding techniques and Precision, Recall and F-score as their evaluation metrics. They compared the results of their proposed models like CNN, LSTM and attention LSTM etc. Navonil *et al* [35] proposed a novel learning scheme. In their work, they trained a classifier for both sarcasm detection and as well as sentiment analysis on a single neural network with the help of multi-task learning. Their proposed network smoothest the interaction between two tasks. By doing this, it can help to enhance the performance of both tasks. They used Gated Recurrent Units (GRU) as their training model. Important steps in their work as follows: Input representation, Sentence-level word representation, Attention network, Sentiment classification and sarcasm classification. Their dataset consists of 994 samples, in which each one consisted of a labelled text snippet with the sentiment and sarcasm tag. For the

comparison between their model and baseline models, they selected stand-alone sarcasm classifier, stand-alone sentiment classifier and CNN-based method. They claimed from their work result that their model showed a slightly better result than the state-of-the-art models.

Santiago Castro *et al* [8] suggested that including multimodal cues can enhance sarcasm detection. They proposed a new dataset, MUStARD (Multi-Modal Sarcasm Detection Dataset). MUStARD includes the audio-visual utterance from famous TV shows. These audio-visual utterances are annotated with sarcasm labels and the context of historical utterance. They worked on different baseline models and showed the importance of multimodal models compared to the unimodal variants of the multimodal models. To create a dataset, they selected sarcastic and non-sarcastic videos. To gather sarcastic videos, they mainly depended on YouTube. They used different keywords like sarcasm in TV shows, Friends sarcasm, sarcasm 101 and Chandler sarcasm. They also collected videos from a famous TV show known as, The big bang theory. To collect non- sarcastic videos, they collected 400 videos from multimodal emotion recognition dataset (MELD). Their dataset consisted of 6421 videos. They did manual annotation on these video segments. They extracted three feature types from the dataset. Text features, Video features, and Speech features. They used BERT in the dataset to represent the textual utterance. They used Librosa, which is a speech processing library, to extract speech features. For extracting video features, they used ResNet-152 image classification model. They used SVM as their main baseline model. They integrated context information and speaker information and used as additional input for their model. From the comparison of results with their model and the baseline model, they suggested the multimodality is important for sarcasm detection. Liuuan Liu *et al* [33] proposed a new deep neural network method for sarcasm detection known as A2Text-Net. They integrated the auxiliary variables like POS, emoji, punctuations and numerals etc., to enhance the performance of sarcasm detection classifier. Their A2Text-Net model combined different auxiliary data with the word embedding output. A2Text-Net has three layers. Hypothesis layer, Feature processing layer, and neural network layer. The main goal of hypothesis layer is to finalize from the given auxiliary variables as appropriate to add to the text. The word embedding is to convert text data into vector form and to convert text data into a structured manner. The input from the first two layers is connected and fed into the neural network layer. They claimed from their experiments that the proposed A2Text-Net improves the performance of classification. They used a back-propagation deep neural network in the neural network layer. They used four public datasets to test the performance of A2Text-Net model. The datasets are News headlines dataset, two Twitter datasets and Reddit dataset. They used F-score, Recall and Precision as their evaluation metrics. The researchers used Logistic regression, SVM, and LSTM etc., as their base models for comparison. They concluded from their test results comparison that their proposed model performed better than baseline models. Mandar and Dangi *et al* [23] proposed a pre-training model based on BERT. They named their model as SpanBERT. They mainly used two different approaches compared to BERT. Rather than masking random tokens, they masked contiguous random spars. They trained these span boundary representations to anticipate the whole content of the masked span. They trained their model on different datasets like SQuAD 1.1, SQuAD 2.0 and GLUE etc. They used BERT as the baseline model for comparison. They claimed based on their work that the spanBERT model outperforms all BERT baseline model on different NLP tasks.

Mikhail Khodak *et al* [25] introduced a large corpus for sarcasm detection research. Their proposed model is known as SARC (Self-Annotated Reddit Corpus). SARC corpus consists of 1.3 million sarcastic statements. SARC has both self-annotated and unbalanced labels. Each sarcastic example in SARC corpus has author, context and topic information. This dataset is a very useful resource for sarcasm detection research. Joshi *et al* [22] worked on the importance of word embedding approaches in sarcasm detection. They studied four different types of word embedding techniques. They studied Glove, word2vec, LSA and Dependency weights. They concluded from their work the usage of word embedding features can increase the performance of classification models and hence improve the performance of the model. Aniruddha *et al* [15] proposed a semantic-based neural network model for sarcasm detection. Their proposed model consisted of a CNN module which is followed by LSTM module and DNN (Deep neural network) module. They compared the performance of their proposed model with Recursive-SVM model. Their dataset contains 18k sarcastic tweets and 21k non-sarcastic tweets. From comparing the test results of their proposed model with the results of Recursive-SVM model, they claimed that their proposed model outperformed the Recursive-SVM model.

## 4. Different Approaches for Sarcasm Detection

In past years, different studies have been carried out on sarcasm detection. We can generally classify sarcasm detection approach to four categories. They are, i) Rule-based approach, ii) Lexicon-based approach, iii) Machine learning-based approach, iv) Deep learning-based approach. Fig 1 shows the general classification of sarcasm detection approaches.

### *4.1. Rule-based approach*

In this approach, sarcasm identification is done based on certain rules or evidences. These rules played as the indicators to capture sarcasm. The rule-based approach used different properties of sentences like semantic, syntactic and lexical etc., to identify different patterns in phrases or uses indicators like hashtags. Tony Veale *et al* [49] proposed an approach to find sarcasm in similes. A simile is an approach to compare two things directly. For example, "She is an angel". In their proposed approach, based on rules, each simile is classified as sarcastic and non-sarcastic using the google search results. Researchers Diana Maynard and Mark A. Greenwood [36] uses the hashtag approach to identify sarcasm from tweets. They studied the effect of sarcasm in sentiment analysis. They created a hashtag tokenizer for GATE. GATE is a framework which helps users to produce and place different resources and language engineering components robustly. Santhosh Kumar Bharti *et al* [5] proposed two approaches to identify sarcasm from the Twitter data. The first approach is PBLGA (Parsing-Based Lexicon Generation Algorithm) and the second approach is to identify sarcasm based on the existence of interjection words. Ellen Rilof *et al* [43] proposed an approach to detect sarcasm by finding the instances where positive sentiment contradicts with a negative situation. For this purpose, they proposed a bootstrapping algorithm. Following are the different types of rule-based methods.

#### *4.1.1. Lexical method*

In the lexical method, different lexical features like verb, noun and adjective etc., are considered. Mohd Suhairi *et al* [47] used n-gram method to extract lexical features from the text. They suggested that the lexical features are one of the common features using in the field of NLP.

#### *4.1.2. Syntactic method*

Syntax means the study of the formation and the structure of a sentence. Also, we can say that syntax is a set of rules based on which a sentence is correctly organized. The syntactic method mainly focused on the POS (Parts-of-Speech) of a sentence. The main difference between syntax and semantics is that the semantics is about the meaning of a sentence and syntax handles the structure of a sentence. Mohd Suhairi *et al* [47] used syntactic features in their work.

#### *4.1.3. Semantic method*

Semantics is a better understanding of the meaning of a language. It also learns the different meaning of words. A single word can have different meaning based on the context. For example, consider the word 'Bat'. Based on the context either it means the animal or sports equipment. The semantic method is one of the commonly using rule-based approaches due to its efficiency in identifying sarcasm. Bharti *et al* [5] used a semantic-based approach for their research work.

### *4.2. Lexicon-based approach*

The main concept of the lexicon-based approach is to use different opinion words to represent different sentiments. A lexicon is a collection of words or a dictionary of a language. There are mainly two types of lexicon-based approaches are there. Corpus-based approach and dictionary-based approach.

#### *4.2.1. Corpus-based approach*

This approach helps to identify the opinion words which dependent on the context. These opinion words depend on syntactic patterns. Ellen Rilof *et al* [43] used a corpus-based approach for their research work. In their work, they identified sarcasm based on when positive sentiments contradict with the situation. For example, "I love being ignored" sentence has the positive sentiment 'love' followed by the part 'being ignored' which depicts a negative situation. WordNet can be used to find word semantic orientation. WordNet is a huge database of English words in which the words are related to each other by its semantics.

#### *4.2.2. Dictionary-based approach*

In the dictionary-based approach, opinion words are gathered manually. This collection of opinion words can be further elaborated by including the synonyms and antonyms of these opinion words which depends on the context. Machine learning and deep learning approaches are discussed in section VI.E.

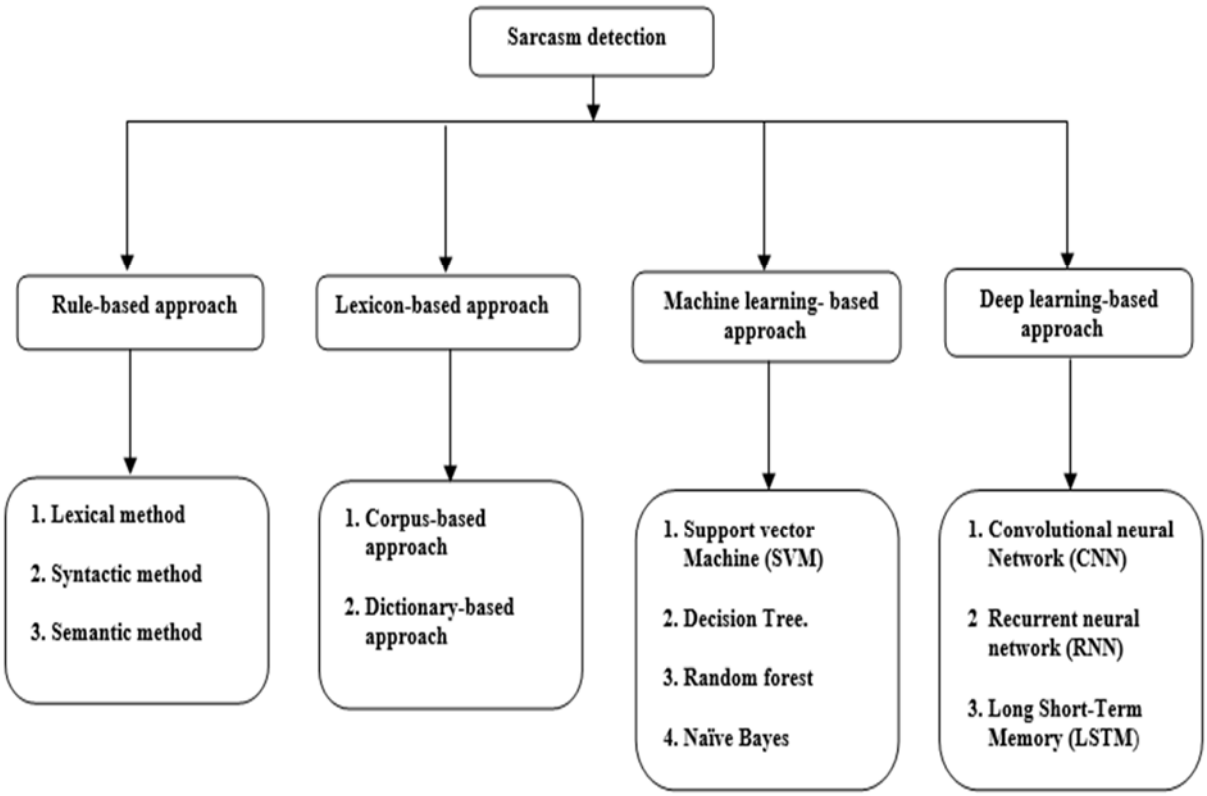


Fig 1. Different sarcasm detection approaches

## 5. General Feature Sets for Sarcasm Detection

Features are one of the most important attributes of any NLP tasks including sentiment analysis and sarcasm detection. Different classification algorithms have to train on well-defined features to predict the more accurate output. A general classification of feature sets using

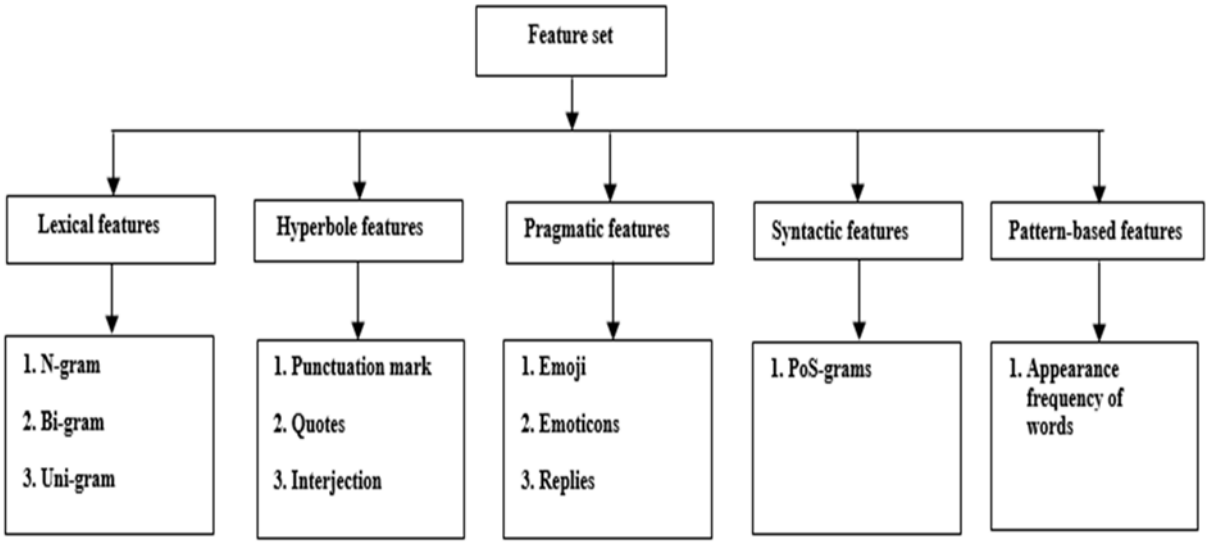


Fig.2 Different feature sets of sarcasm*5.1. Lexical features*

One of the most important features in sarcasm detection is lexical features. Lexical features mainly include the properties of text. Examples for lexical features are n-gram, bi-gram and uni-gram. N-gram is a method used in NLP. We generally say that n-gram is a combination of words. 'N' in n-gram indicates the number of the word combination. For example, if n=1, then it is known as uni-gram. If n=2, it is known as bi-gram and so on. In uni=gram, each word considers separately or we can say that each word is a 'gram'. Consider the following sentence, "I am a student". In uni-gram representation, this sentence will split into each word or each gram. "I", "am", "a", and "student". In bi-gram representation, every two neighbour words will be considered together. "I am", "am a", and "a student".

### *5.2. Hyperbole features*

Hyperbole is a way of expressing feelings in an exaggerated manner. Intensifiers and punctuations are examples of hyperbole. Also, other properties of text like interjections and quotes are considered as hyperbole. Presence of hyperbole in a sentence is also a strong indication for the presence of sarcasm. Researchers also frequently used hyperbole features to increase the accuracy of the classification model for sarcasm detection. For example, "WoW!!! I failed again in exam??!!".

### *5.3. Pragmatic features*

Emoticons, smileys and emojis are examples of pragmatic features. As we know from the literature review section, several researchers used pragmatic features as the key features to train the classification models to detect sarcasm. This is due to the presence of pragmatic features like emojis etc., shows a strong indication of sarcasm.

### *5.4. Syntactic features*

The main example of syntactic features is PoS-gram or Part-of-Speech gram. PoS-gram is a group of tokens or words which are annotated with Part-of-Speech tags. Morphosyntax is the study of syntactic and morphological characteristics of a text. In linguistic, morphology means the study of word-forming patterns in a language. Syntactic features usually consist of repeated order of morphosyntactic pattern.

### *5.5. Pattern-based features*

Some words appear in text data with high frequency compared to other words in that text data. These high-frequency words are used in a pattern-based feature approach. Repeated or frequently appearing word patterns indicates the presence of sarcasm.

## 6. General Architecture of Sarcasm Detection

The general architecture of sarcasm detection shows in figure 3. The main steps in sarcasm detection are Data collection, Pre-processing of data, Feature extraction, Detection of sarcasm using classifiers and Result calculation.

### *6.1. Data collection*

Data collection or data acquisition is the first step in sarcasm detection. Generally, there are two ways to collect data for sarcasm detection purpose. One way is to use APIs. API (Application Program Interface) is an application that uses to interacts with software. API helps users to interact with applications. For example, users can access Twitter data by using Twitter API. Users can download tweets by mentioning specific hashtag. In sarcasm detection, a commonly used hashtag is #sarcasm. Similarly, Facebook Provides Graph API, so researchers can directly access sites like Facebook, Twitter, Amazon and Reddit etc., to collect the data, which allowed by sites. Another method is to use already available datasets for sarcasm detection, for example, datasets like SARC (Self-Annotated Reddit Corpus), IAC (Internet Argument Corpus), SQuAD (Stanford Question Answering dataset), MUStARD (Multi-Modal Sarcasm Detection Dataset), SemEval (Semantic Evaluation) dataset etc.

### *6.2. Data pre-processing*

After collecting the data, data should be pre-processed before giving as input to the classification model. This is because the collected data may be unstructured and contain noises. So, data pre-processing is an important task in sarcasm detection. Different types of data pre-processing techniques are, stop-word removal, tokenization of data, stemming and lemmatization.

#### *6.2.1. Stop word removal*

Stop words are trivial or common words in any languages which have less important for model training. For example, in the English language, an, a, the, is etc., are the common stop words. These words are removed during pre-processing due to its less importance. NLTK (Natural language tool kit) is a library which is using in Python for stop word removal.

#### *6.2.2. Tokenization of words*

Tokenization means that converting a sentence into words or tokens. NLTK library is using for tokenization of sentences in Python. For example, consider the sentence, 'I am happy'. By using 'word_tokenize' function from NLTK, the sentence is converted into three words or tokens as I, am, happy. In the same way, a paragraph can be converted to sentences by using sent_tokenize function of NLTK.

*6.2.3. Stemming and Lemmatization*

Stemming is converting a word into its root form. Lemmatization is also the same process with a slight difference. For example, consider the following three words. Going, goes and gone. By using stemming, these three words are converted to its root form 'go'. Lemmatization converts the words into more meaningful representation. For example, if we use lemmatization to following three words, finally, final and finalized, we will get the root word as 'final'. But if stemming is applied to those words, the root word getting is, 'fina'. So, for some words, the stemming process may not give meaningful root word representation. Lemmatization takes more time than stemming. For both stemming and lemmatization, we can use the NLTK library. For example, 'PorterStemmer' can be used for stemming purpose and for lemmatization, 'WordNet Lemmatization' can be used.

***6.3. Feature extraction***

Feature extraction is the process to extract features and based on these features, forms a feature representation which is appropriate for the NLP task or the classification model. There are different features extraction methods are there like Bag of Words, TF-IDF and word embeddings techniques like word2vec and Glove.

*6.3.1. Bag of words*

when training deep learning or machine learning models, the input text data should be converted into a vector form. This process is called vectorization. Consider the following sentence, 'Sarcasm is a difficult task'. After applying pre-processing techniques, which we mentioned in the previous section, the sentence converted to three words. Sarcasm, difficult and task. After this process, a histogram is created for these words. Histogram means the frequency representation of words in a given sentence or document. After that, this histogram representation should be converted into vector form. Bag of Words is one the process used to convert words to vector form. So, each sentence converted into vector form representation. The problem with Bag of Words is that the order of occurrence of words is lost. Means, different features can have the same vector representations.

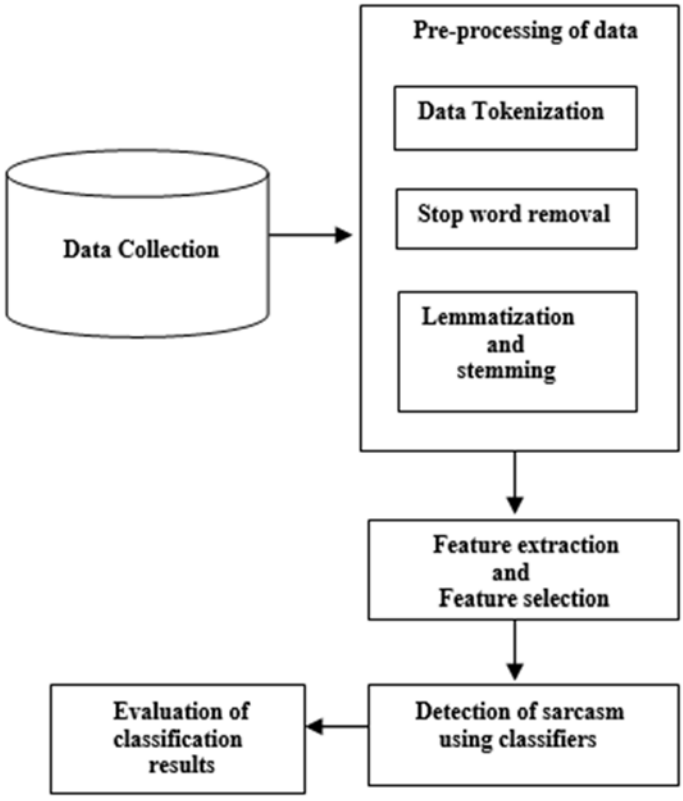


Figure 3. Flow diagram of Sarcasm detection

*6.3.2. TF-IDF (Term Frequency-Inverse Document Frequency)*

TF-IDF converts words to vector form. TF (Term Frequency) and IDF (Inverse Document Frequency). TF calculates how frequently a word is present in the given document and IDF finds the words which appear in very few sentences in the given document. So, together. TF-IDF calculate a score which finds the frequency of each word present in a given document. The disadvantage of TF-IDF is that it can't give semantic information.

*6.3.3. Word embedding*

Word embedding also converts words into vector form. The main advantage of Word embedding is that it gives the semantic information of words and as well as it is able to find the relation between different words presented in the given document. Word2vec and Glove are the famous word embedding techniques.

*(i) Word2vec*

Word2vec is introduced by Tomas Mikolov. Word2vec is a neural network with two layers. It is a very effective word embedding to approach to learn from raw text data. In word2vec representation, similar words mapped to nearby values. In word2vec word embedding, the connection between different words and semantic information is maintained. For example, queen and women are represented as related words in word2vec.

*(ii) Glove*

Glove (Global Vectors for Word Representations) is another famous word embedding method. Glove method based on a technique known as the matrix factorization. Based on the given words, Glove produces a matrix based on co-occurrence information. The glove can find how frequently a word in some context is present in large data corpus. We can say that Glove is a count-based model and word2vec is a predictive model. After creating a matrix for co-occurrence information, it converts the matrix to a lower-dimension matrix by doing the factorization on that matrix.

### *6.4. Feature selection*

In general terms, we can say that the feature selection is a method to select the most important feature sets to improve the performance of classification models. Main feature selection methods are chi-square ($\chi^2$) and mutual information (MI) techniques.

*6.4.1. Chi-square ($\chi^2$)*

In this method, the chi-square value between the target and each one of the features is calculated. The features with better values of chi-square are selected for training the model.

Because the higher value of chi-square indicates the more dependency between the feature and the expected value. To analyze the values, the chi-square method uses a contingency table. Chi-square testing mainly uses to find statistical dependence or independence between categorical variables. The first step in the chi-square test is to define a hypothesis. The contingency table is formed after defining the hypothesis. Before finding the chi-square values, expected values should be determined. Based on chi-square values, the null hypothesis is either accepted or rejected.

*6.4.1. Mutual information (MI)*

Mutual information between two random variables means the degree of dependence or mutual dependence among these two random variables. Mutual information methods help to model mutually dependent variables. We can reduce the dimensionality by using mutual information feature selection method. Mutual information can also find non-linear relationships between random variables. If the value of mutual information is zero, it means that the two random variables are statistically independent.

### *6.5. Sarcasm classification techniques*

Sarcasm detection is done by using different classifiers like machine learning algorithms and deep learning algorithms. Recent research in sarcasm detection mainly using deep learning approach or combination of different deep learning approaches with machine learning models.

*6.5.1. Machine learning models.*

Machine learning (ML) comes under the Artificial Intelligence (AI). Machine learning methods gives the ability to a system to learn, adapt and improve from experience. In simple words, we can define that, machine learning provides the ability to the system to learn without the help of human assistance. Supervised, unsupervised, reinforcement and semi-supervised learnings are the types of machine learning. SVM, Decision tree, Random forest and Naïve Bayes are examples of machine learning algorithms. The machine learning algorithm process can be divided into two phases. Training phase and Testing phase. Training phase helps the machine learning algorithm to learn. In the test phase, the evaluation of the model is tested using some evaluation metrics.

*(i) SVM (Support vector machine)*

SVM is a supervised machine learning algorithm. SVM applies to both regression problem and classification problems. The main purpose of SVM is to form the best decision boundary which can separate n-dimensional space to a different class. So, when a new data point comes, the model can correctly analyze and place this new data point into the appropriate class. The main strength of SVM is a hyperplane. The hyperplane is the most appropriate or perfect decision boundary. Support vectors are the extreme points or vectors which helps to form hyperplane. There are two types of SVM. Linear SVM and Non- Linear SVM. Linear SVM can linearly separate data into two classes by the help of a single straight line. Non- Linear SVM can separate data in a non-linear

manner. It means that given data points can't be classified by only using a single straight line. Characteristics of hyperplane depend on the features in the dataset. SVM performs well within a space of high dimensionality. But SVM takes more training time.

*(ii) Decision Tree.*

Decision Tree can also be used for both regression problem and classification problems. It is a supervised machine learning model. This model represented as a tree structure and hence named as a decision tree. In this tree-structure, internal nodes, branches and leaf nodes are present. The internal nodes are the features of a dataset. Decision rules represented as the branches and output represented as the leaf node. There are two nodes in the decision tree. Decision node and leaf node. Decisions are making by decision node. That's why the decision node has many branches. A leaf node is for showing output based on the decision. This node has no branches. Features from the dataset are the factors for making decisions.

Classification and regression tree algorithm (CART) is using to create a decision tree. The main advantage of a decision tree is that it tries to mimic the way human thinking when it comes to making a decision.

*(iii) Random forest*

Random forest is a supervised machine learning algorithm. This machine learning approach is suitable for both regression problems and classification problems. The main concept behind the random forest algorithm is ensemble-based learning. In an ensemble learning approach, different classifiers are combined to find the solution for a complex problem and to enhance the performance of the model. The random forest consists of multiple numbers of decision trees. Random forest analyses the result from each decision tree present in the model and then apply the majority voting method for predicting the output. Due to having many numbers of decision trees, the random forest can be able to solve the overfitting problem and to enhance the accuracy in predicting the correct output. Compared to other supervised algorithm models, Random forest usually takes less time to predict the output. It is also a good method to train larger datasets and predicting result with higher accuracy.

(*iv) Naïve Bayes*

Naïve Bayes classifier algorithm is a supervised machine learning algorithm which is using to find a solution for classification tasks. Main application area of Naïve Bayes is the text classification. This algorithm can do a quick prediction. It is a simple, effective and one of the fast machine learning models. It anticipates the results based on probability. So, we can say that Naïve Bayes is a probabilistic classifier. Bayes' law is a rule-based approach to find the probability of a given hypothesis based on prior knowledge. The model is known as Naïve because it presumes that the presence of some features is entirely independent of the presence or occurrence of other features. This algorithm works based on Bayes' law. So, this model is known as Naïve Bayes. One of the advantages of Naïve Bayes is that it can apply on Binary classification and also in the multi-class classification. . Naïve Bayes is one of the preferred choices in the area of textual classification tasks.

*6.5.2. Deep learning models.*

Deep learning is the branch of machine learning. Deep learning invented based on the function and structure of the human brain working system. The basic building block concept of deep learning is an artificial neural network. Deep learning has more neurons, complex methods of connecting layers and has the ability to automatic feature extraction. Deep learning neural network has a large number of parameters. The main types of deep learning methods are CNN (Convolutional neural network), RNN (Recurrent neural network), recursive neural networks and unsupervised pre-training networks. Most of the deep learning models use neural network architecture. The name deep learning comes because of the large number of hidden layers present in the deep learning model. Generally, a normal neural network has 2-3 hidden layers. But a deep neural network can have several hidden layers.

*(i) CNN (Convolutional neural network).*

CNN is a deep learning-based algorithm. Main application area of CNN is image classification, facial recognition and identifying objects etc. consider the scenario that doing image classification using CNN. CNN process the input image and classifies it based on features from image data. Before classification, each input image will undergo through several convolutional layers in CNN. Convolutional layer, Max pooling layer, Flattening layer and Full connection layer are the different layers in the CNN. Convolutional layer extracts the feature set from the input images. Convolutional layer creates "Feature Map". Feature map is created by input images and feature detector. Filters or feature vector is a matrix of order 3*3 or 5*5 etc. When an image is too large, the pooling layer reduces the parameters. Down sampling technique is for dimensionality reduction. Max pooling is one of the examples for downsampling technique. In the input matrix, the pixel shift is applied. The number of this pixel shift on the input matrix is known as a stride. Sometimes, the input image not perfectly fit in the filter. That's why padding technique is used. Padding means including picture matrix with zeros, this is known as zero padding. We can summarize the working of CNN as follows. The input image is fed into the convolutional layer. After selecting parameters, filters with strides are applied. If necessary, padding is used. CNN uses ReLU (Rectified Linear Unit)

activation function. After performing convolution to the image, ReLU function is applied to the matrix. Next step is to apply pooling to reduce the size of dimensionality. The numbers of convolutional layers should be decided with analyzing the working of the model. Before giving output to the fully connected layer, the output should be flattened. To output the class, the activation function is needed. After applying the activation function, the next step is to classify the images.

*(ii) RNN (Recurrent neural network).*

RNN is a type of neural network. In RNN, the input giving to the current step of processing is the output coming from the previous step. One advantage of RNN is that it has memory. It can use its internal state as memory and can process a sequence of inputs. This advantage makes RNN do tasks like generating text and language modelling. In RNN, every input is related to other inputs. Another advantage of RNN is that it can work with convolution layer and to create an effective neighbourhood for pixel. RNN can remember the context of input during the training process. Gradient vanishing and exploding problems are the disadvantages of RNN. LSTM model overcomes these issues.

*(iii) LSTM (Long Short-Term Memory).*

LSTM is an enhanced version of RNN. LSTM overcomes the vanishing gradient problem. LSTM uses backpropagation technique to train the model. LSTM has three gates. Input gate, Forget gate and Output gate. Input gate decides to choose values from inputs and use selected values to update the memory. Sigmoid and tanh activation functions are used in LSTM. As the name indicates, forget gate decides to discard which details from the model. The output is decided by input gate and the memory.

### *6.6. Evaluation of classification results*

The final procedure in sarcasm detection is to evaluate the results of the classification model. This process is done using different evaluation metrics. Important evaluation metrics are confusion metrics, accuracy, precision, recall and f-measure.

#### *6.6.1. Confusion matrix*

For binary classification, a confusion matrix consists of four entities. True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). Two types of instances are there in the binary classification confusion matrix. Actual instance and predicted instance. The confusion matrix compares the actual instances with the instances or values predicted by the classifier model. Figure 4 shows the 2 * 2 confusion matrix.

*(i) True Positive (TP)*

If the predicted instance matches with the actual instance, we call it the True Positive. Means, the actual instance was positive and the classification model accurately predicted the positive value.

*(ii) True Negative (TN)*

The actual instance was negative value and the classification model accurately predicted the negative value for the predicted instance.

*(iii) False Positive (FP)*

False Positive is also known as the Type I error. The value of the actual instance was negative. But the classification model wrongly predicted a positive value for the predicted instance.

Table 1. Comparison of different classification models of sarcasm detection.

| Author | Classification techniques | Datasets | Results |
|---|---|---|---|
| Yufeng dieo *et al* [10] | Bi-LSTM | IAC V2 (Internet argument corpus) | Precision = 70.1 |
| Himani and Vaibhav *et al* [46] | Hierarchical BERT based model | Reddit and Twitter data set | Twitter F-score = 0.74<br>Reddit F-score = 0.639 |
| Edoardo *et al* [45] | Bi-LSTM and Multi-Layer perceptron (MLP) | SARC data set which consists of sarcastic comments from Reddit site | F-score = 0.763 |
| Adithya *et al* [3] | BERT based model | Reddit and Twitter data set | Twitter F-score = 0.75<br>Reddit F-score = 0.621 |
| Sayed and Agarwal *et al* [44] | LSTM-RNN | Twitter data set | Accuracy = 85.23 |
| Vaishvi *et al* [19] | SVM classifier model | Dataset consisted of sarcastic and regular news headlines | Accuracy = 78.82 |
| Jens *et al* [31] | Ensemble-based approach | Reddit and Twitter data set | Twitter F-score = 0.74<br>Reddit F-score = 0.67 |
| Nan XU *et al* [52] | D&R Net (Decomposition and relation network) | Data set of images which contain the text. | Accuracy = 84.02 |
| Nirmala *et al* [38] | An unsupervised probabilistic relational framework | Twitter data set | F-score = 0.82 |
| Hongliang and Zhong *et al* [39] | SAWs (Self-Attention of Weighted Snippets). | Twitter data set<br>IAC | Accuracy = 83.21 |
| Hankyol Lee *et al* [30] | Contextual Response Augmentation (CRA) model | Reddit and Twitter data set | Precision = 0.34 |
| Jacob *et al* [9] | Bidirectional encoder representations from Transformers (BERT). | SQuAD v1.1 (The Stanford Question Answering Dataset) | F-score = 92.4 |
| Lu Ren *et al* [42] | CNN | IAC | Precision = 87.40 |

*(iv) False Negative (FN)*

False Negative is also known as the Type 2 error. The value of the actual instance was positive. But the classification model wrongly predicted a negative value for the predicted instance.

Using the value of TP, TN, FP and FN, we can calculate the precision, recall, accuracy and f-measure.

$$\textit{(v) Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (2)$$

$$\textit{(vi) Precision} = \frac{(TP)}{(TP + FP)} \quad (3)$$

$$\textit{(vii) Recall} = \frac{(TP)}{(TP + FN)} \quad (4)$$

$$\textit{(viii)}\ \text{F-measure} = \frac{(2 * Recall * Precision)}{(Recall + Precision)} \quad (5)$$

Table 2. Confusion matrix

| Predicted instances | Actual instances | |
|---|---|---|
| Positive | TP | FP |
| Negative | FN | TN |

## 7. Challenges in Sarcasm Detection.

Sarcasm is one of the difficult tasks in sentiment analysis. That is why sarcasm detection is known as the Achilles' heel of sentiment analysis. Researchers already proved from their work that correctly detecting sarcasm can enhance the sentiment analysis performance.

(i) Main challenge in sarcasm detection is the ambiguous nature of sarcastic words. Understanding this ambiguity and decrypting its inner meaning is one of the major challenges.

(ii) So many features have to consider in sarcasm detection when compared to sentiment analysis. Perfectly extracting these all features and train the classification model appropriately is also

a challenging work.

(iii) Sarcastic data has so many noises. Finding sarcasm from short and noisy text is difficult and at the same time the results from noise analysis don't add much help to sarcasm detection,

(iv) Choosing appropriate classification model is also an important factor. A classification model worked well on Twitter dataset may not be work effectively on Reddit dataset and vice versa.

(v) Another challenge is the quality of the dataset. Usage of different slang and informal language using in covey text messages is difficult to understand. The dataset which doesn't have hashtags can make it harder to train the classification model in the expected way.

(vi) Compared to sarcasm detection from speech and textual data, finding sarcasm in textual data is more difficult. Speech features like rhythm, unique tones, body gestures, eye-contact and facial expression etc., are not present in textual features.

## 8. Conclusion and Future Work

Sarcasm detection is one of the interesting topics in sentiment analysis. Due to the ambiguity and complex nature of sarcasm, it is a difficult task to detect sarcasm more precisely from the dataset. This challenge is one among the reason that researchers are attracted to sarcasm detection work. Due to the importance of social media sites, sentiment analysis and sarcasm detection also gained popularity. This paper discussed the recently conducted research work on sarcasm detection. Also briefly discussed the challenges in sarcasm detection and different classification models used in different works. From the analysis of recent works done on sarcasm, the researchers of this paper concluded that deep learning-based models and machine learning-based models are mainly using in the research area of sarcasm detection. Deep learning approaches like LSTM and CNN become more popular than other classification models in the area of sarcasm detection. Also. Reddit dataset s*et al*so frequently used in new research works. Most of the previous works are done in Twitter datasets only. So, exploring more and more datasets from different areas is also a good way of approaching the sarcasm detection problem. We would like to give some suggestions as for future work in sarcasm detection based on our literature study. i) Datasets using in sarcasm detection is limited to certain areas like Twitter, Reddit etc. Using different datasets from other areas can be useful in future research. ii) Most sarcasm detection works done on the English language. Considering other languages as for future work is a considerable idea. iii) Most of the works are based on textual data. Multi-modal sarcasm detection can increase the possibility to explore new ideas about sarcasm detection. iv) Exploring new features which can give the idea about the context of sarcasm also seems to a better idea. v) Ensemble-based approaches can increase the accuracy of sarcasm detection.

## References

[1] Agrawal, A., An, A. and Papagelis, M., 2020, July. Leveraging Transitions of Emotions for Sarcasm Detection. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 1-4).
[2] Ashok, D.M., Ghanshyam, A.N., Salim, S.S., Mazahir, D.B. and Thakare, B.S., 2020, June. Sarcasm Detection using Genetic Optimization on LSTM with CNN. In 2020 International Conference for Emerging Technology (INCET) (pp. 1-4). IEEE.
[3] Avvaru, A., Vobilisetty, S. and Mamidi, R., 2020, July. Detecting Sarcasm in Conversation Context Using Transformer-Based Models. In Proceedings of the Second Workshop on Figurative Language Processing (pp. 98-103).
[4] Baruah, A., Das, K., Barbhuiya, F. and Dey, K., 2020, July. Context-aware sarcasm detection using BERT. In Proceedings of the Second Workshop on Figurative Language Processing (pp. 1373-1380).
[5] Bharti, S.K., Babu, K.S. and Jena, S.K., 2015, August. Parsing-based sarcasm sentiment recognition in twitter data. In 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (pp. 1373-1380). IEEE.
[6] Camp, E., 2012. Sarcasm, pretense, and the semantics/pragmatics distinction. Noûs, 46(4), pp.587-634.
[7] Campbell, J.D. and Katz, A.N., 2012. Are there necessary conditions for inducing a sense of sarcastic irony?. Discourse Processes, 49(6), pp.459-480.
[8] Castro, S., Hazarika, D., Pérez-Rosas, V., Zimmermann, R., Mihalcea, R. and Poria, S., 2019. Towards Multimodal Sarcasm Detection (An _Obviously_ Perfect Paper). arXiv preprint arXiv:1906.01815.
[9] Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
[10] Diao, Y., Lin, H., Yang, L., Fan, X., Chu, Y., Xu, K. and Wu, D., 2020. A Multi-Dimension Question Answering Network for Sarcasm Detection. IEEE Access, 8, pp.135152-135161.
[11] Dong, X., Li, C. and Choi, J.D., 2020. Transformer-based context-aware sarcasm detection in conversation threads from social media. arXiv preprint arXiv:2005.11424.
[12] Dutta, S., Mehta, A. and Bandyopadhyay, S.K., A Novel Integrated Framework for Sarcasm Detection in Social Platform.
[13] Eisterhold, J., Attardo, S. and Boxer, D., 2006. Reactions to irony in discourse: evidence for the least disruption principle. Journal of Pragmatics, 38(8), pp.1239-1256.
[14] Farha, I.A. and Magdy, W., 2020, May. From Arabic Sentiment Analysis to Sarcasm Detection: The ArSarcasm Dataset. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection (pp. 32-39).
[15] Ghosh, A. and Veale, T., 2016, June. Fracking sarcasm using neural network. In Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis (pp. 161-169).
[16] Gupta, R., Kumar, J. and Agrawal, H., 2020, May. A Statistical Approach for Sarcasm Detection Using Twitter Data. In 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 633-638). IEEE.
[17] Ivanko, S.L. and Pexman, P.M., 2003. Context incongruity and irony processing. Discourse processes, 35(3), pp.241-279.
[18] Jaiswal, N., 2020, July. Neural sarcasm detection using conversation context. In Proceedings of the Second Workshop on Figurative Language Processing (pp. 77-82).
[19] Jariwala, V.P., Optimal Feature Extraction based Machine Learning Approach for Sarcasm Type Detection in News Headlines. International Journal of Computer Applications, 975, p.8887.
[20] Javdan, S. and Minaei-Bidgoli, B., 2020, July. Applying Transformers and Aspect-based Sentiment Analysis approaches on Sarcasm Detection. In Proceedings of the Second Workshop on Figurative Language Processing (pp. 67-71).
[21] Jena, A.K., Sinha, A. and Agarwal, R., 2020, July. C-net: Contextual network for sarcasm detection. In Proceedings of the Second Workshop on Figurative Language Processing (pp. 61-66).
[22] Joshi, A., Tripathi, V., Patel, K., Bhattacharyya, P. and Carman, M., 2016. Are word embedding-based features useful for sarcasm detection?. arXiv preprint arXiv:1610.00883.
[23] Joshi, M., Chen, D., Liu, Y., Weld, D.S., Zettlemoyer, L. and Levy, O., 2020. Spanbert: Improving pre-training by representing and predicting spans. Transactions of the Association for Computational Linguistics, 8, pp.64-77.
[24] Khatri, A., 2020. Sarcasm detection in tweets with BERT and GloVe embeddings. arXiv preprint arXiv:2006.11512.
[25] Khodak, M., Saunshi, N. and Vodrahalli, K., 2017. A large self-annotated corpus for sarcasm. arXiv preprint arXiv:1704.05579.
[26] Khotijah, S., Tirtawangsa, J. and Suryani, A.A., 2020, July. Using LSTM for Context Based Approach of Sarcasm Detection in Twitter. In Proceedings of the 11th International Conference on Advances in Information Technology (pp. 1-7).
[27] Kumar, A. and Anand, V., 2020, July. Transformers on sarcasm detection with context. In Proceedings of the Second Workshop on Figurative Language Processing (pp. 88-92).
[28] Kumar, A., Narapareddy, V.T., Srikanth, V.A., Malapati, A. and Neti, L.B.M., 2020. Sarcasm Detection Using Multi-Head Attention Based Bidirectional LSTM. IEEE Access, 8, pp.6388-6397.
[29] Kumar, A., Sangwan, S.R., Arora, A., Nayyar, A. and Abdel-Basset, M., 2019. Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network. IEEE Access, 7, pp.23319-23328.
[30] Lee, H., Yu, Y. and Kim, G., 2020. Augmenting Data for Sarcasm Detection with Unlabeled Conversation Context. arXiv preprint arXiv:2006.06259.
[31] Lemmens, J., Burtenshaw, B., Lotfi, E., Markov, I. and Daelemans, W., 2020, July. Sarcasm detection using an ensemble approach. In Proceedings of the Second Workshop on Figurative Language Processing (pp. 264-269).
[32] Lin, C.Z., Ptaszynski, M., Masui, F., Leliwa, G. and Wroczynski, M., 2020. A Study in Practical Solutions to Sarcasm Detection with Machine Learning and Knowledge Engineering Techniques. In AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering (1).
[33] Liu, L., Priestley, J.L., Zhou, Y., Ray, H.E. and Han, M., 2019, December. A2Text-Net: A Novel Deep Neural Network for Sarcasm Detection. In 2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI) (pp. 118-126). IEEE.
[34] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V., 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.
[35] Majumder, N., Poria, S., Peng, H., Chhaya, N., Cambria, E. and Gelbukh, A., 2019. Sentiment and sarcasm classification with multitask learning. IEEE Intelligent Systems, 34(3), pp.38-43.
[36] Maynard, D.G. and Greenwood, M.A., 2014, March. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In LREC 2014 Proceedings. ELRA.
[37] Misra, R. and Arora, P., 2019. Sarcasm detection using hybrid neural network. arXiv preprint arXiv:1908.07414.
[38] Nimala, K., Jebakumar, R. and Saravanan, M., 2020. Sentiment topic sarcasm mixture model to distinguish sarcasm prevalent topics based on the sentiment bearing words in the tweets. Journal of Ambient Intelligence and Humanized Computing, pp.1-10.
[39] Pan, H., Lin, Z., Fu, P. and Wang, W., Modeling the Incongruity between Sentence Snippets for Sarcasm Detection.

[40] Pant, K. and Dadu, T., 2020. Sarcasm Detection using Context Separators in Online Discourse. arXiv preprint arXiv:2006.00850.
[41] Pawar, N. and Bhingarkar, S., 2020, June. Machine Learning based Sarcasm Detection on Twitter Data. In 2020 5th International Conference on Communication and Electronics Systems (ICCES) (pp. 957-961). IEEE.
[42] Ren, L., Xu, B., Lin, H., Liu, X. and Yang, L., 2020. Sarcasm Detection with Sentiment Semantics Enhanced Multi-level Memory Network. Neurocomputing.
[43] Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N. and Huang, R., 2013, October. Sarcasm as contrast between a positive sentiment and negative situation. In Proceedings of the 2013 conference on empirical methods in natural language processing (pp. 704-714).
[44] Salim, S.S., Ghanshyam, A.N., Ashok, D.M., Mazahir, D.B. and Thakare, B.S., 2020, June. Deep LSTM-RNN with Word Embedding for Sarcasm Detection on Twitter. In 2020 International Conference for Emerging Technology (INCET) (pp. 1-4). IEEE.
[45] Savini, E. and Caragea, C., 2020, April. A Multi-Task Learning Approach to Sarcasm Detection (Student Abstract). In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 10, pp. 13907-13908).
[46] Srivastava, H., Varshney, V., Kumari, S. and Srivastava, S., 2020, July. A novel hierarchical BERT architecture for sarcasm detection. In Proceedings of the Second Workshop on Figurative Language Processing (pp. 93-97).
[47] Suhaimin, M.S.M., Hijazi, M.H.A., Alfred, R. and Coenen, F., 2017, May. Natural language processing based features for sarcasm detection: An investigation using bilingual social media texts. In 2017 8th International Conference on Information Technology (ICIT) (pp. 703-709). IEEE.
[48] Thenmozhi, D., 2020, July. Sarcasm Identification and Detection in Conversion Context using BERT. In Proceedings of the Second Workshop on Figurative Language Processing (pp. 72-76).
[49] Veale, T. and Hao, Y., 2010, August. Detecting ironic intent in creative comparisons. In ECAI (Vol. 215, pp. 765-770).
[50] Wilson, D., 2006. The pragmatics of verbal irony: Echo or pretence?. Lingua, 116(10), pp.1722-1743.
[51] Xiong, T., Zhang, P., Zhu, H. and Yang, Y., 2019, May. Sarcasm Detection with Self-matching Networks and Low-rank Bilinear Pooling. In The World Wide Web Conference (pp. 2115-2124).
[52] Xu, N., Zeng, Z. and Mao, W., 2020, July. Reasoning with Multimodal Sarcastic Tweets via Modeling Cross-Modality Contrast and Semantic Association. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 3777-3786).
[53] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R. and Le, Q.V., 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In Advances in neural information processing systems (pp. 5753-5763).