Journal of Autonomous Intelligence (2024) Volume 7 Issue 4
doi: 10.32629/jai.v7i4.800

ORIGINAL RESEARCH ARTICLE

Detection of sarcasm in tweets using hybrid machine learning method

Bellamkonda Rajani[1], Sameer Saxena[1], Billakurthi Suresh Kumar[2,*]

[1] *Amity Institute of Information Technology, Amity University Rajasthan, Jaipur 303002, India*

[2] *School of Computer Science and Engineering, Sanjay Ghodawat University, Kolhapur, Maharashtra 416118, India*

*** Corresponding author:** Billakurthi Suresh Kumar, sureshkumarbillakurthi@gmail.com

ABSTRACT

When one wants to express the contrary of what they mean, especially when insulting someone, sarcasm is used. One of the challenges associated with the control and management of content on social networking sites such as Twitter and other social media sites, is the recognition of sarcasm. Because sarcasm is purposefully conveyed through ambiguous word choice, even for humans it can be challenging to identify. Existing methods for automatically detecting sarcasm rely heavily on lexical as well as linguistic clues the majority of the time. On the other hand, these methods have not yielded a large or even a minor enhancement in terms of the correctness of the mood. The purpose of this study is to increase the accuracy of sentiment analysis by proposing a system that is both reliable and effective at identifying sarcasm. In this investigation, three different types of features are focussed: lexical, sarcastic, and context features. These feature sets are utilised in the process of categorising tweets as either sarcastic or not sarcastic. This research presents a sarcastic feature set coupled with an efficient hybrid machine learning approach, which ultimately results in improved accuracy. The results of the experiments reveal that the recommended hybrid machine learning method achieves 97.3% accuracy for sarcastic feature sets which is better when compared to existing machine learning techniques namely k-nearest neighbor, random forest, support vector machine and decision tree for the selection of the appropriate features.

Keywords: sarcasm detection; sentiment; traditional machine learning classifiers; hybrid machine learning method

ARTICLE INFO

Received: 7 July 2023
Accepted: 4 September 2023
Available online: 4 February 2024

1. Introduction

A sophisticated style of communication known as sarcasm occurs when the speaker delivers the complete opposite of what they intended to say. As a result of this, sarcasm is frequently confused with irony or humor. This is because the most common applications of sarcasm are to criticise, insult, or entertain. Its use has even been linked to rage in the workplace and the development of linguistic skills in youngsters. Furthermore, it is not practical to convey visual or tone signals when engaging in online conversations. The ability to identify sarcasm is a crucial component of opinion mining and has a wide range of applications, including those in the fields of health, defense, and sales[1]. A number of organizations and businesses have expressed an interest in doing research on the data included in tweets in order to learn what people think about well-known brands, political events, or films. The content of Twitter is significantly increased because to the daily addition of billions of tweets to the platform[2,3].

However, the informal nature of the language used in tweeting and social media platforms makes it significantly more difficult to grasp the feelings of other users and to conduct analysis on those emotions using machine learning and deep learning techniques[4–7]. Furthermore, sarcasm makes sentiment analysis more difficult than it has to be and leads to incorrect categorization of people's points of view using various feature extraction and classification methodologies[6,7]. As a result, it causes a reduction in the accuracy of the sentiment classification. People will use sarcasm in a statement or while conversing to show disrespect for another person or to mock them using ensemble learning techniques[8]. People will use upbeat language to cover up their negative emotions. In recent times, sarcasm and irony are quite prevalent in social media, despite the fact that it can be difficult to recognise them. The definition of the word "ironic" refers to a situation in which the motive behind the real composition and the speech are identical. For instance, the satirical statement "I love to study on vacations!" displays a contradiction between the term "love" and the precise statement "on vacations."

The researchers have an interest in detecting sarcasm in social media text, particularly in tweets[9]. A serious situation arises when there is rapid growth in the number of tweets. It is also referred as opinion minin using machine learning algorithms[10], which is the process of deducing the attitude or viewpoint of a speaker from the opinions of other people. Since a few years ago, numerous scholars have shifted their attention on sentimental analysis as a primary area of interest, notably in the field of social network. By offering a collection of techniques and processes, the methodologies and algorithms of machine learning offer a new path for sentiment analysis, in particular the identification of sarcasm[11,12].

Utilizing a wide variety of feature extraction strategies, the previously conducted research attempted to identify sarcastic tone in tweets. A survey of the available research on irony identification reveals that the existing methods suffer from two significant flaws[13,14]. Firstly, the properties of the words in depiction that are dependent on sarcasm, lexicon, and context are not taken into consideration. As a result, numerous words can have equivalent representations in vector space. The second issue is the sparsity of the dataset within the vector form, due to the fact that each statement has a word limit. Due to these problems, the majority of the training features set vectors may be sparse when the models are being trained[15]. This is because some words may only appear in the testing set, while others may not appear in either the training or testing sets at any point. As a result, the purpose of this study is to construct a reliable model for sarcasm detection that is dependent on the feature set training dataset. For the purpose of sarcasm detection in text mining evaluation, an effective feature set and hybrid machine learning classifier model have been recommended as the most effective methods for achieving considerably better precision[16].

The recognition of sarcasm in posts on twitter improves the effectiveness of sentiment analysis, which is the primary contribution of this research. To accomplish this, a number of machine learning classification algorithms and a variety of features were utilised in order to determine the best classification algorithm with major features. Several experiments are performed using three feature sets namely sarcasm, context and lexical features. It is observed that the accuracy of sarcasm detection using proposed Hybrid Machine Learning (HML) is 97.3% which is far better as compared to other traditional Machine Learning (ML) classification models.

The remaining portions of this article are organised as follows: The review of related literature is discussed in Section 2, the framework of proposed HML for detecting sarcasm is described in Section 3. The results of the experiments are presented in detail in Section 4, and the overall discussion is wrapped up in Section 5.

2. Literature survey

According to Shrikhande et al.[1] sarcasm is a crucial feature of communication; nonetheless, it is challenging for people, much less computers, to recognise sarcasm in written or spoken language. It appears that sarcasm is used frequently in the titles of newspapers in an effort to attract the interest of their readers.

Although, the majority of the time, readers have trouble seeing the sarcasm in the headlines. As a result, they form the incorrect assumption that the news item in question is true, which they then communicate to their circle of friends, coworkers, and other individuals. Therefore, it is more necessary than ever to have a system that is capable of instantly and dependably detecting sarcasm. Therefore, sarcasm detectors are created through the use of neural networks, and an effort is made to comprehend how a machine understands the patterns of sarcastic speech. The project's input comprises of patterns that are either sarcastic or non-sarcastic, depending on how they are categorised. These sequences were taken from two independent datasets, one including news stories and the other having comment from social media. The classifiers' degrees of precision are investigated and ranked. The model that was proposed achieves high performance and is capable of precisely categorising phrases as either sarcastic or not sarcastic.

Python was used in the creation of a programme by Zanchak et al.[2] that can detect sarcasm in written text. A dataset obtained from the Kaggle website, which compiles headlines of news articles from web pages based in the United States. The decision was made to use logistic regression and a bayesian classifier, along with simple tokenization and vectors retrieval depending on TF-IDF, as well as the building of neural networks using LSTM and GRU. For the purpose of weight extraction, the Glove and Word2Vec techniques were chosen. The neural network models with weights generated by the use of the Glove approach demonstrated the highest level of accuracy: 80.5

Liu et al.[3] utilised a brand new deep neural network called A2Text-Net to replicate the face-to-face speech. This network includes auxiliary variables to boost classification performance. These variables include proper punctuation, parts of speech (POS), numerals, emoji, and so on. The findings of the experiment give proof that our A2Text-net technique achieves superior classification efficiency when compared to both traditional machine-learning and deep learning techniques.

Sarcasm identification is very helpful to many different online platforms, as per Liu[4], which helps actualize sentiment analysis, subjectivity analysis, and advertisement push. As a result, the level of research into the identification of sarcasm has recently reached its highest point. In recent years, a number of research have utilised deep learning in order to handle sarcasm detection challenges. This is possible due to the advancement of deep learning as well as the effective application of media networks in NLP. These studies proposed novel models that were based on methods (such as convolutional neural networks and Converters), and these systems were put through a significant amount of testing through a wide variety of tests to investigate how well they performed. When compared to the traditional model, the model that was developed as a result of this research is continuously refined. This paper gives an overview of some earlier work that was based on Convolutional Neural Networks (CNN), transformers, and other networks. It also provides a list of important factors, incorporates some research frameworks and strategies, and provides an assessment of classifier index. The work that was done to produce this article will serve as a useful reference for the future researchers.

According to Kanakam and Nayak[5] it is exceedingly challenging for human beings to detect sarcasm in written language. Because of this, researchers planned to automatically detect sarcasm using computer simulations. People today frequently utilise various forms of social media to share their thoughts, ideas, and reactions to the posts made by others. At this point in time, it seems to be the norm for everyone to make use of sarcastic language in their everyday conversations, either directly or indirectly. When someone uses sarcasm in a conversation, they are imparting a positive connotation to words that they are saying. In other terms, it is the use of sarcasm to mock, communicate scorn, or say the opposite of what you truly mean to say. Irony is used to do all of these things. People are critical while using positive language. In this article, a variety of strategies for identifying sarcasm in online social media were analysed and discussed. An evaluation of each strategy has been carried out, and its advantages and disadvantages, as well as its level of effectiveness in regards to precision, recall, and accuracy, have been outlined.

Sentiment analysis is a task that examines an opinion presented to determine if it is positive, negative, or neutral; however, the existence of sarcasm changes the valence of the text. This is according to Godara and Aron[6]. The detection process can be carried out using any number of different machine learning techniques. Several processes, including data set gathering, feature extraction, and classification, are utilised during the process of sarcasm detection. The accuracy of the classifier can also be improved by the use of a strategy that reduces the number of features it considers. On the Twitter dataset, the effectiveness of several different classifiers, including Naive Bayes (NB), Principal Component Analysis (PCA), support vector machine (SVM), k-nearest neighbour and k-means clustering, was evaluated and compared. The integration of k-mean, principal component analysis, and support vector machine delivers great accuracy in terms of performance when contrasted to other classification models.

The research conducted by Razali et al.[7] focuses on recognizing sarcasm in tweets by combining deep learning extracted characteristics with contextually generated information. Following the extraction of a feature set from the architecture of a CNN, it is then mixed with feature sets that have been painstakingly built. The corresponding contextual explanations served as the inspiration for the creation of these hand-crafted feature sets. Every single one of these feature sets was developed expressly for the purpose of spotting sarcasm in text. The goal is to identify the characteristics that are the most desirable. Even when utilised separately, certain sets can be used successfully without any problems. If the other sets aren't combined with anything, they won't actually amount to anything noteworthy. The experiments produced favourable findings in the areas of accuracy, specificity, recall, and F1-measure. In order to facilitate comparisons, the various combinations of characteristics are categorised using a selection of machine learning strategies. It has been determined that the Logistic Regression technique is the most effective classification method for this problem. In addition, as supplementary items of data, the results are compared to more recent works as well as the effectiveness of every feature set are shown. In addition, as supplementary items of data, the results are compared to more recent works as well as the effectiveness of every feature set are shown.

According to Venkatesh and Vishwas[8] Sarcasm is a form of humour that reacts to a circumstance by expressing the contrary of what you intend in order to ridicule fun of others, and the act of stating the contrary of what you intend in order to poke fun of someone else. The use of a sarcasm reorganisation strategy can be highly helpful in improving the accuracy of computerized sentiment analysis data obtained from tweeting and social networking sites. The study of internet users' reported sentiments and perspectives in a certain group, in addition to their classification and grouping, is what's meant to be referred to as "sentiment analysis," and the phrase has been given its own name for the process. Identifying sarcasm is one of the most challenging issues that arises in the field of sentiment analysis. The categorization of sarcastic phrase types is a challenging endeavour. For the purpose of identifying real-time sarcasm on Twitter, this study employs two hybrid machine learning techniques, particularly regarding Stacked Generalization and Boosting ensemble techniques with SVM, RF and KNN as base classification method and Logistic Regression (LR) as Meta classification models. Specifically, the work focuses on the use of Support Vector Machine (SVM), Random Forest (RF), and KNN.

The sarcasm detection process and techniques are discussed in Bhat and Jha[9], along with an evaluation of the obtained results on a variety of models and datasets. The process of sarcasm identification is somewhat analogous to that of sentiment classification, which uses mathematical models to illustrate and classify the emotional tone of a passage of text or phrase in order to establish whether or not it contains sarcasm. Identification of sarcasm has been carried out in recent years on a variety of online platforms, including Twitter databases, Reddit data, the SARC dataset, and many others. The primary focus of this research is on the various deep learning and machine learning strategies to sarcasm identification, such as SVM, CNN and LSTM models that have been employed for sarcasm detection in latest studies. In the course of this research, a diverse range of approaches to sarcasm identification were investigated. At the conclusion, it examined the various methods for detecting sarcasm and contrasted and compared them based on how accurately they performed.

According to research conducted by Gamova et al.[10], internet trolls have a significant influence on other users, which makes it difficult for people to use the internet comfortably. The development of a model for spotting inflammatory and fraudulent comments is the goal of this project. We searched for science publications that discussed the detection of trolls manually and used those articles as one of the eligibility criteria for informative comments. In order to accomplish this objective, it has developed two artificial neural networks: one for sarcasm definition and another for sentiment analysis identification. The output of the programme is a dataset consisting of troll posts and false remarks, as well as statistics and diagrams of characterization.

A multi-head attention-based bidirectional long-short memory (MHA-BiLSTM) architecture was presented by Kumar et al.[11] in order to identify sarcastic remarks within a given corpus. The results of the study indicate that a multi-head attention mechanism improves the performance of BiLSTM, and that it provides better performance than feature-rich SVM models as a consequence of this improvement.

Sarcasm, which is a stinging style of presenting the information, can also be included in the text, as per Rao and Sindhu[12]. The first thing that needs to be done is choosing the dataset. Amazon datasets are accessed in order to retrieve the dataset. The next step is the preparation of the data, which involves polarity detection, separating, and tokenization as well as text categorization of the data. After that, the process of feature extraction is carried out, which consists of the following steps: n-gram, term frequency, and inverse document frequency. The classification algorithms such as SVM, K Nearest Neighbors, and RF are applied. After that, the correctness of the parameter is used to evaluate the computation of the results.

A context-based feature strategy for sarcasm identification is proposed by Eke et al.[13], who use the deep learning model, the BERT model, and traditional machine learning in order to solve the problems described in the previous paragraph. Benchmark datasets from Twitter as well as Internet Argument Corpus, version 2 were used in the classification process carried out by the three learning models. The first model makes use of an embedding-based depiction by means of a deep learning approach with bidirectional long short term memory (Bi-LSTM), which is a subtype of Recurrent Neural Network (RNN), and by employing Global Vector representation (GloVe) for the development of word embedding and context acquiring knowledge. The next model is constructed utilising a pre-trained Bidirectional Encoder interpretation and Transformer. It is based on the first model (BERT). The third model, on the other hand, is focused on feature fusion rather than traditional machine learning. This model combines BERT features, sentiment-related characteristics, syntactic features, and GloVe embedding attributes. In order to determine how successful this method is, a number of different evaluation experiments are conducted. The evaluation of the method on two Twitter data sets, on the other hand, achieved the greatest possible precision of 98.5% and 98.0%, correspondingly. On the contrary hand, the IAC-v2 dataset attained the maximum precision of 81.2%, which demonstrates the relevance of the suggested method in comparison to the baseline methodologies for sarcasm analysis.

Using data from Twitter, Pawar and Bhingarkar[14] suggested a pattern-based method for identifying sarcasm in written communication. It is hypothesised that there are four sets of traits that can identify tweets as either sarcastic or non-sarcastic based on the amount of particular sarcasm they include. The suggested feature sets are being investigated, and their extra cost classes are being evaluated.

For the purpose of multi-modal sarcasm detection, Sangwan et al.[15] suggest an efficient solution that is founded on deep learning and makes use of data that is both text and visual. The strategy that has been suggested makes use of recurrent neural networks and seeks to anticipate outcomes by capitalising on the interaction that exists between the various types of inputs. According to the findings of the experiments, the integration of visual modalities appears to play a significant influence in the enhancement of performance.

Nguyen et al.[16] proposes a thorough coding guide to categorise sarcastic content that is shared on social media by including terminology from well-known dictionaries that are written in the English language. This guide is used to assign labels to the tweets that were collected after two antilock down rallies in Michigan. From the tweets that have been categorised, a set of features that encompass situational, lexical, social, sentimental, and auxiliary factors are retrieved. Many typical machine learning models are trained with the help of these features. These models have an F1-score of 0.83 and are able to distinguish between sarcastic and non-sarcastic tweets. Without using any guiding keywords, this method has the potential to achieve competitive accuracy in sarcasm detection using data aggregated from two distinct contexts. This makes the method very potential. It is suggested that sarcastic inclinations and their manifestations may be inherent to individuals, may supersede the frame of reference, and as a result, intended to guide for building handheld classifiers for automatic identification that are agnostic to the context. Significance scores suggest that non-contextual attributes contribute approximately 57% to the detection.

3. Research methodology

Figure 1 provides an illustration of the structure of the proposed methodology, which is divided into three distinct units: the data preprocessing unit, the features extraction unit, and the classification unit. The data is initially taken from the Twitter dataset, which is available on Kaggle. In the second step of the process, the data is pre-processed by removing noise, eliminating stop words, stemming, lemmatization, removing punctuation etc. The pre-processed data is split up into two distinct parts, a training dataset (consisting of 80% of the dataset) and a testing dataset (consisting of 20% of the dataset). In the third step, features are extracted and selected before being passed over to the classification unit to be used as an input. When classifying tweets into sarcastic and non-sarcastic categories, a number of different machine learning classifiers as well as proposed HML technique have been used. In the end, a comparative evaluation of the performance of the proposed HML classifier and the standard ML classifier is carried out utilising performance metrics such as accuracy, precision, recall, and f-score. Consider the **Figure 1**, which illustrates how well the suggested HML approach can identify sarcasm in textual content. The detailed description of each unit of the framework is discussed as follows.

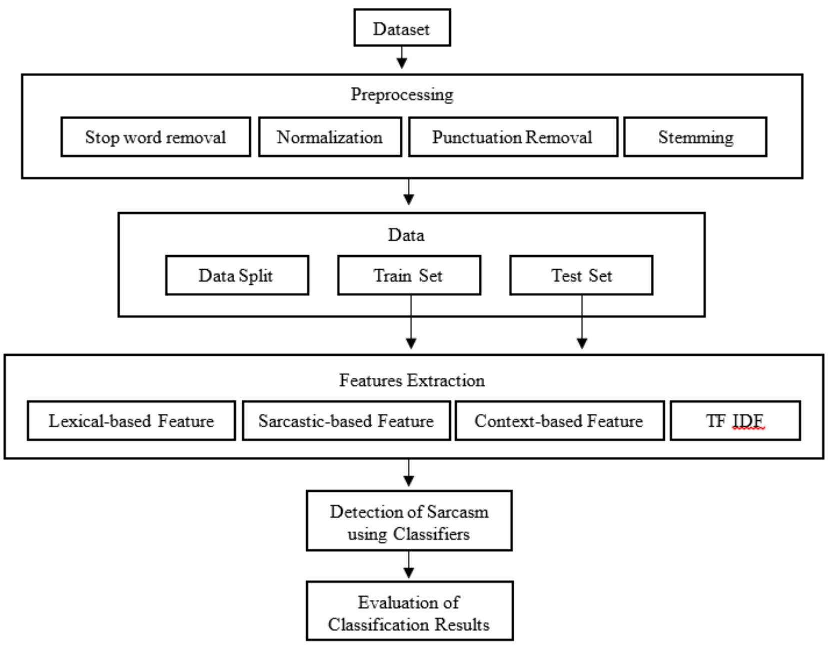

Figure 1. Framework of proposed HML for detecting sarcasm.

- **Data Collection:** The real time twitter and Kaggle combined dataset, which can be found on kaggle, is the source of the data that is gathered. This online sarcasm corpus is open to the public and includes a dataset with three columns: the first column is an index of tweets, the second column contains the text content, and the third column has labels. The below **Table 1** describes a details description of dataset including URL and attributes information. It is made up of 3834 labels, evenly distributed between sarcastic labels provided by (1) and non-sarcastic labels provided by (0).

Table 1. Dataset description.

URL	Text	class
https://www.huffingtonpost.com/entry/versace-black-code_us_5861fbefe4b0de3a08f600d5	Former versace store clerk sues over secret ‘black code’ for minority shoppers	is_sarcastic: 0
https://politics.theonion.com/boehner-just-wants-wife-to-listen-not-come-up-with-alt-1819574302	Boehner just wants wife to listen, not come up with alternative debt-reduction ideas	is_sarcastic: 1

- **Data Pre-processing:** The noise that is added to a dataset after it has been obtained via social networks is one of the drawbacks associated with this method of data collection. In this section, the data is pre-processed using a range of different methods, which are explained as follows:
- **Tokenization:** Tokenization refers to the process of slicing longer chunks of text, such as sentences or words, into more manageable chunks known as tokens. These tokens can include symbols, words, and statements that are useful on their own. This tokenization lessens the amount of white space visible in the character set of the text document.
- **Removal of Stop Word Removal:** The three most common types of stop words are articles (the, an, and a) and prepositions (to, if, of, and). Stop words are also combined with another type of commonly used word. All of these, in point of fact, do not have any impact on determining whether or not sentences in text data include sarcasm. As a result, it is eliminated before beginning the process of analysing the dataset.
- **Removal of noise:** Twitter-saved keywords, single-word tweets, and other types of tweets should have superfluous symbols, non-ASCII characters, and number newlines removed from them.
- **Stemming:** The method of stemming reduces the total amount of keyword terms while simultaneously improving the categorization display performance of the specific keywords that are acquired from the different variations of that keyword. E.g., ‘Cluster’ can stem from ‘Clustering’.
- **Removal of Punctuation:** Since punctuation creates noise for textual data sentences, which are represented by !@#$^%^&*()_)_:”?>{}these are removed before extracting features from the dataset. Certain punctuation marks, such the question mark or exclamation sign, can sometimes be seen as an indication of irony (sarcasm).
- **Data Splitting:** In this stage of the procedure, the pre-processed dataset is split into two parts: the training dataset and the testing dataset, with an 80:20 split ratio. This data has been preprocessed before being sent on to the feature extraction unit as an input.
- **Feature Extraction:** In this step, three feature set namely lexicon, sarcastic and context feature are used to classify the textual data into sarcasm or non-sarcasm. The detail description of each feature set is described as follows:
- **Sarcastic based feature set:** In order to arrive at a conclusive response to a problem, people frequently turn to the use of convoluted or complicated words that incorporate obscure terminology in an effort to render them unreadable to the reader or listener. People intentionally try to conceal their genuine emotions or viewpoints when they use sarcasm to prevent the uncomfortable conversation, even though they may appear to be laughing. Sometimes, people will try to relocate the sarcasm data by using punctuation, such as a repeated ellipsis, punctuation mark. Because of this, the total number of ellipses,

punctuation mark that are repeated has been determined. Additionally, interjections that come after the words ‘oh-oh-oh’, ‘he-he-he,’ ‘awh,’ shh,’ ‘eww,’ ‘amazing,’ ‘aww,’ ‘kiddo,’ ‘uh,’ ‘hmm’. Someone managed to get people to communicate their feelings in a variety of different ways, which is what sets sarcastic tweets apart from the dataset. Words like “looovvvve,” and “gooooddddd,” as well as duplicated letters in groups of more than two, are likely indicators of sarcastic tweets. Likewise, the repetition of vowels serves the same purpose. Therefore, incorporate multiple repeated vowels and letter components into the collection of tweets that you have. Because of this, the feature is retrieved prior to the process of deleting duplicated letters from words. Some people, on the other side, will demonstrate their derision by using all upper case letters (for instance, FANTASTIC and WONDERFUL) as well as sections of the term written in all upper case letters (for instance, fANtastic and woNderFul). Count the amount of laughs, paying special attention to expressions such as “lmao,” “omg,” “lhh,” “joking”, “hehehehe,” “wtf,” and “aaah”. Repeated number quotations are likewise counted towards the total number of tweets, just like repeated letters are.

- **Lexical based feature set:** The tweets are strongly impacted by the adjectives, verbs, nouns, and adverbs that are used. It is counted independently according to the wording of each individual tweet. In addition, amplification words such as horribly, bloodily, startlingly, totally, stupidly, absurdly, enormously, terribly, astonishingly, etc. When someone wants to express negative thoughts during a positive vice versa or intensifiers tweet, they will sometimes exaggerate the tweet. As a result of this, a count is taken of the number of both negative and positive intensifiers that are contained within each tweet. In conclusion, the opinions of all tweets are computed, revealing the tweets’ polarity in its simplest form.
- **Context-based feature set:** People have a tendency to communicate their actions by using a variety of hashtags as well as posting remarks on various online platforms. It’s possible that the connection between the viewers and the users, followed by a direct communicative context, is significant for boosting the precision of the sarcasm prediction. Sarcasm identification on a social network was accomplished with the help of a context-based model that was used to post-level irony identification. Additional context can be provided for sarcasm identification by using quantitative information, such as the author’s real tweets.
- **Term Frequency-Inverse Document Frequency (TF-IDF):** Inverse Document Frequency, often known as IDF, is a method that, when combined with term frequency, is applied to a data collection in order to reduce the significance of the influence of the text terms that are used most frequently. IDF gives the greatest importance to a word based on whether it appears in a text document with either a low or high frequency of occurrence.
- **Classification and evaluation:** In this step, proposed HML method and standard machine learning classifiers are used to classify the tweet into sarcasm or non-sarcasm. Various experiments are performed to evaluate the performance of proposed HML using performance metrics namely accuracy, precision, recall and f-score.

4. Algorithm design

To generate an effective training matrix, this method employs both lemma and rule-based features. The dependency relations discussed in section B are used to extract rule-based features, and the multivariate filter approach is used to pick Lemma features. To create a single training matrix, the training matrices of both types of features are concatenated. At the time of testing, both lemma and rule-based features are extracted, and the aspect category label is chosen using the same testing process as the base system. Combining rule-based features with lemma features improves the effectiveness of aspect category prediction, according to this experiment.

The proposed system follows the steps of Algorithm 1 and Algorithm 2; preprocessing, feature extraction, feature selection, generation of training matrix and evaluation of system performance using test dataset in a sequential manner.

Algorithm 1 Hybrid feature selection using Document Frequency (DF) + Hybrid Weighted + Co-relation Coefficient (CC) method

Input: Number of feature in TF set, Weighted TF Denominator = 3, CC Threshold th.
Output: Generate train matrix each term which contains 0-1 weight for each aspect category.
Step 1: Fset = All Features of TF
Step 2: for each (F in Fset)
    Frequenct(F) ← TF[f]
    if (Frequenct(F) ≥ TFDenominator)
    weightedList.add(F)
end for
Step 3: for each (t in weightedList)
    aspectTA ← aspect[t]
    Owen_Count ← t.occurances(aspectTA)
    Other_count ← t.occurances(! aspectTA)
$$aspect\_weight(t) = \frac{\mathrm{Own}_{[\mathrm{Count}]}}{\mathrm{Other}_{[\mathrm{count}]}+\mathrm{Own}_{[\mathrm{Count}]}}$$
    Update weight for t ← aspect_weight(t)
End for
Generate weighted matrix according to calculated weight for respective term which fulfils the various threshold criteria for each category.
Step 4: To calculate the correlation coefficient weight for each term using below X and Y matrix, here $X(term)[]$ represents the matrix of current term t with all n aspect index as well as $Y(term)[]$ illustrates next terms t+1 index matrix, the both matrix must having a same index, before calculation of correlation weight of term t with t + 1.
$$X(term)[] = \sum_{m=1}^{n}(.\,aspect[m] \ldots\ldots .\,aspect[n].count)$$
$$Y(term+1)[] = \sum_{k=1}^{n}(.\,aspect[k] \ldots\ldots .\,aspect[n].count)$$
Step 5: Apply coefficient correlation on X and Y matrix using below formula
$$C0weight\ [term] = \frac{\mathrm{n}\ (\sum \mathrm{X}[]\mathrm{Y}[]) - (\sum \mathrm{X}[]) * (\sum \mathrm{Y}[])}{\sqrt{[\mathrm{n} \sum \mathrm{X}^2 - (\sum \mathrm{X})^2]}\ \sqrt{[\mathrm{n} \sum \mathrm{Y}^2 - (\sum \mathrm{Y})^2]}}$$
Step 6: To select the best feature from available feature set after correlation coefficient weight calculation using given threshold
if ($C0weight\ [term] \leq th$)
FinalFeatureset ← *{Term, Aspect}*
Step 7: Generate final matrix for each term called correlation coefficient matrix

Algorithm 2 Proposed HML Classification Algorithm

Input: Weighted coefficient correlation matrix for each selected term M, Test dataset test_data
Output: aspect label prediction for each instance
Step 1: for each t in M
Step 2: read all index M [i…n] values
Step 3: if ($M[i] \geq 0.0$)
    Newlable ← Convert label 1 for respective t
    MN[] ← Newlable[i……n]
End for
Step 4: generate updated binary matrix MN[]
Step 5: input and preprocess test_data
$$Lemmas[] = \sum_{k=0}^{n} input[k \ldots .n].\{Stopword, Lemmitization\}$$
Step 6: for each ($s\ in\ Lemmas$)
    T[] ← s.split(tokens)
For each (term t in T)
$$f(x) = t\ ||\ \sum_{n=1}^{m}(MN[n].values) \text{ if(exist)}$$
Step 7: calculate mean for each aspect category based on $f(x)$
Step 8: calculate belief for all categories
Step 9: Return highest belief category as aspect for test instance.
End for

5. Result and discussion

Various experiments are performed to evaluate and compare the performance analysis of standard ML classifiers and proposed HML using performance metrics namely precision, accuracy, recall and f1-score. Consider the following **Table 2** which illustrates the comparative analysis of proposed HML and standard ML classifiers.

Table 2. Comparative analysis of performance of proposed HML and standard ML and DL classifiers.

Algorithms	Precision	Recall	F1-score	Accuracy
k-nearest neighbour	73.13	43.45	54.45	61.62
Decision Tree	92.63	92.12	89.52	85.32
Random Forest	92.80	91.85	90.35	80.96
Support Vector Machine	93.89	92.95	91.38	90.64
RNN	92.40	91.50	91.80	92.10
LSTM	94.09	93.90	94.20	94.70
PNN	93.55	92.60	93.10	93.05
Proposed HML	95.03	93.76	92.88	97.30

Above **Table 2** shows that the performance of proposed HML is better as compared to standard machine learning as well as deep learning classification models.

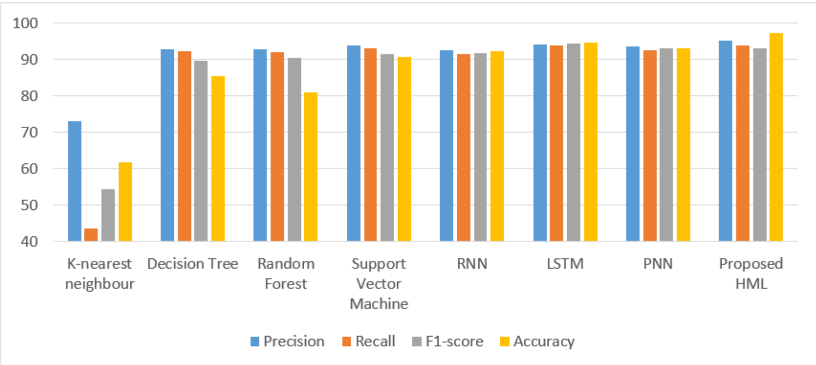

Figure 2. Performance comparison of proposed HML and standard ML classifiers.

Consider the above graph which is depicted in **Figure 2**, it shows that the performance accuracy, precision, recall and f1-score of proposed HML is better as compared to standard ML classification models.

Table 3. Performance analysis of accuracy using different feature selection techniques.

Algorithms	Lexical features	Sarcastic features	Content features	Hybrid
k-nearest neighbour	55.9	61.05	58.9	61.62
Decision Tree	83.25	85.36	80.44	85.32
Random Forest	75.32	80.39	78.12	80.96
Support Vector Machine	88.35	90.57	89.63	90.64
RNN	90.30	91.70	88.80	92.10
LSTM	91.35	92.50	89.95	94.70
PNN	92.80	95.90	94.80	93.05
Proposed HML	96.28	97.18	96.9	97.30

Consider the **Table 3** and **Figure 3**, where performance analysis of accuracy of 3 feature sets namely lexical, sarcastic and context feature is performed. Thus, it is observed from the experimental findings that the performance of proposed hybrid machine learning classification model for sarcastic feature set as compared to other machine learning techniques namely k-nearest neighbour, random forest, support vector machine and decision tree.

6. Limitation of work

The research describes an sarcasm detection using machine learning, while promising, is not without its limitations. Here are some of the key challenges and limitations associated with proposed system:

- **Lack of clear indicators:** Sarcasm often relies on context, tone, and subtleties that can be difficult to capture in written text. Many sarcastic statements might not have explicit markers, making it challenging for machine learning models to accurately identify them.
- **Context dependency:** Sarcasm often relies on understanding the broader context of a conversation, including previous statements and shared background knowledge. Machine learning models might struggle to capture the nuances of context, leading to misinterpretations.
- **Cultural differences:** Sarcasm can vary significantly across cultures and languages. A model trained on one cultural context might not perform well when applied to another, as it may not understand the cultural cues that indicate sarcasm.
- **Sarcasm types and variability:** Sarcasm can take various forms, including hyperbole, understatement, irony, and more. Detecting all these forms accurately requires a model to understand a wide range of linguistic nuances.

Addressing these limitations requires a combination of improved training data, advanced natural language understanding models, context-aware approaches, and potentially a deeper integration of world knowledge to understand subtle cues that indicate sarcasm.

7. Conclusion and future work

On Twitter, sarcasm is a more developed and nuanced form of irony that may be encountered rather frequently. Finding sarcastic tweets is a crucial step in the text classification process, and doing so has a wide range of repercussions as a result. Due to this, this study analysed several different machine learning algorithms in order to identify the sarcastic statements that can be found on Twitter. An enhanced model for identifying sarcasm in tweets is presented as a potential solution in this research study. In terms of sarcasm identification in tweets, the results indicate that sarcastic features are more dominant than other features. The experimental findings indicate that accuracy is improved when sarcastic-based characteristics are utilised for all classifiers, which was the case throughout this investigation. In addition, the sarcastic-based features of the suggested HML provide the highest level of accuracy, which is approximately 97.3%. In the future, these features can be merged to find out how well different machine learning and deep learning algorithms work.

Author contributions

Conceptualization, BR; methodology, BR; software, BR; validation, BR; formal analysis investigation, BR; resources, BR; data curation, BR; writing—original draft preparation, BR; writing—review and editing, BR; visualization, SS and BSK; supervision, BSK; project administration, BR, SS and BSK. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare no conflict of interest.

References

1. Shrikhande P, Setty V, Sahani DA. Sarcasm detection in newspaper headlines. In: Proceedings of the 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS); 26–28 November 2020; Rupnagar, India. pp. 483–487.
2. Zanchak M, Vysotska V, Albota S. The sarcasm detection in news headlines based on machine learning technology. In: Proceedings of the 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT); 22–25 September 2021; Lviv, Ukraine. pp. 131–137.
3. Liu L, Priestley JL, Zhou Y, et al. A2Text-Net: A novel deep neural network for sarcasm detection. In: Proceedings of the 2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI); 12–14 December 2019; Los Angeles, CA, USA. pp. 118–126.
4. Liu X. Deep learning techniques for sarcasm detection. In: Proceedings of the ICMLCA 2021; 2nd International Conference on Machine Learning and Computer Application; 17–19 December 2021; Shenyang, China. pp. 1–5.
5. Kanakam R, Nayak RK. Sarcasm detection on social networks using machine learning algorithms: A systematic review. In: Proceedings of the 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI); 3–5 June 2021; Tirunelveli, India. pp. 1130–1137.
6. Godara J, Aron R. Support vector machine classifier with principal component analysis and K Mean for sarcasm detection. In: Proceedings of the 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS); 19–20 March 2021; Coimbatore, India. pp. 571–576.
7. Razali MS, Halin AA, Ye L, et al. Sarcasm detection using deep learning with contextual features. *IEEE Access* 2021; 9: 68609–68618. doi: 10.1109/ACCESS.2021.3076789
8. Venkatesh B, Vishwas HN. Real time sarcasm detection on twitter using ensemble methods. In: Proceedings of the 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA); 2–4 September 2021; Coimbatore, India. pp. 1292–1297. doi: 10.1109/ICIRCA51532.2021.9544841
9. Bhat A, Jha GN. Sarcasm detection of textual data on online SocialMedia: A review. In: Proceeding of the 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE); 28–29 April 2022; Greater Noida, India. pp. 1981–1985.
10. Gamova AA, Horoshiy AA, Ivanenko VG. Detection of fake and provokative comments in social network using machine learning. In: Proceedings of the 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus); 27–30 January 2020; St. Petersburg and Moscow, Russia. pp. 309–311.
11. Kumar A, Narapareddy VT, Aditya Srikanth V, et al. Sarcasm detection using multi-head attention based bidirectional LSTM. *IEEE Access* 2020; 8: 6388–6397. doi: 10.1109/ACCESS.2019.2963630
12. Rao MV, Sindhu C. Detection of sarcasm on amazon product reviews using machine learning algorithms under sentiment analysis. In: Proceeding of the 2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET); 25–27 March 2021; Chennai, India. pp. 196–199.
13. Eke CI, Norman AA, Shuib L. Context-based feature technique for sarcasm identification in benchmark datasets using deep learning and BERT model. *IEEE Access* 2021; 9: 48501–48518. doi: 10.1109/ACCESS.2021.3068323
14. Pawar N, Bhingarkar S. Machine learning based sarcasm detection on twitter data. In: Proceedings of the 2020 5th International Conference on Communication and Electronics Systems (ICCES); 10–12 June 2020; Coimbatore, India. pp. 957–961.
15. Sangwan S, Akhtar MS, Behera P, Ekbal A. I didn’t mean what I wrote! Exploring multimodality for sarcasm detection. In: Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN); 19–24 July 2020; Glasgow, UK. pp. 1–8.
16. Nguyen H, Moon J, Paul N, Gokhale SS. Sarcasm detection in politically motivated social media content. In: Proceedings of the 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom); 2021; New York City, NY, USA. pp. 1538–1545.