WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Sarcasm Detection by Using Classifiers and Machine Learning Approaches

[1]Muppudathi R, [2]Lavanya Mathiyalagi A, [3]Jeya Mary S

[1,2,3]Assistant Professor, Computer Science and Engineering, Einstein College of Engineering, Tirunelveli.

ABSTRACT:

Sarcasm, a sophisticated form of irony, is commonly encountered on social networks and micro-blogging platforms, often used for purposes like criticism and mockery. To enhance automatic sentiment analysis of data from these sources, it is essential to detect sarcasm effectively. In this paper, a novel pattern-based approach is proposed for detecting sarcasm on Twitter. By automatically generating rules, the need for costly manual analysis is reduced. The study utilizes a substantial dataset and employs various classifiers, including SVM, Random Forest, Logistic Regression, Decision Tree, Neural Networks, and Naive Bayes, with the goal of achieving high accuracy in identifying sarcastic statements.

Index terms: *Sarcasm, Pattern-based, Classifiers, SVM, Random forest, Logistic Regression, Decision Tree, Neural network, Naïve Bayes*

1. INTRODUCTION

In today's digital age, the internet usage has experienced an unprecedented surge, allowing individuals to freely express their thoughts and emotions on platforms such as Face book and Twitter. Amid these expressions, sarcasm emerges as a distinctive mode of communication, blending humor with subtle criticism. The comprehension of sarcasm involves the collaboration of various cognitive processes within the brain, making it a multifaceted phenomenon. Sarcasm possesses a dual nature, often serving as a source of amusement while also conveying underlying meanings or intentions. Sentiment analysis, especially in the realm of social networks, has garnered significant attention from researchers over recent years. The advent of machine learning methods and algorithms has opened new avenues for sentiment analysis, including the detection of sarcasm, offering a diverse range of effective approaches to explore this intriguing aspect of communication.

1.1 MACHINE LEARNING

Machine learning (ML) could be a variety of computer science that permits software applications to become more accurate at predicting outcomes without being explicitly programmed to try and do so. Machine learning algorithms use historical data as input to predict new output values. Machine learning could be a field of inquiry dedicated to understanding and building methods that ‘learn’, that is, methods that leverage data to boost performance on some set of tasks. It’s seen as a component of AI. Machine learning algorithms build a model supported sample data, referred to as training data, so as to create predictions or decisions without being explicitly programmed to try to so. Machine learning algorithms are utilized in a good form of applications, like in medicine, email filtering, speech recognition, and computer vision, where it’s difficult or unfeasible to develop conventional algorithms to perform the needed tasks. A subset of machine learning is closely associated with computational statistics, which focuses on making predictions using computers, but not all machine learning is statistical learning. Some implementations of machine learning use data and neural networks in a very way that mimics the working of a biological brain. In its application across business problems, machine learning is additionally said as predictive analytics. A core objective of a learner is to generalize from its experience. Generalization during this context is that the ability of a learning machine to perform accurately on new, unseen examples/tasks after having experienced a learning data set. The training examples come from some generally unknown probability distribution (considered representative of the space of occurrences) and also the learner should build a general model about this space that permits it to supply sufficiently accurate predictions in new cases.

1.2 SARCASM DETECTION

The goal of **Sarcasm Detection** is to determine whether a sentence is sarcastic or non-sarcastic. Sarcasm is a type of phenomenon with specific perlocutionary effects on the hearer, such as to break their pattern of expectation. Consequently, correct understanding of sarcasm often requires a deep understanding of multiple sources of information, including the utterance, the conversational context, and, frequently some real world facts.

2. SYSTEM STUDY

2.1 EXISTING SYSTEM

The process of sarcasm detection is essentially a binary classification problem, with the two labels being "sarcastic" and "non-sarcastic". The pre-processed comments at this point were not ready to be used as input in standard machine learning models. Additional data transformation was required.

2.2 PROPOSED SYSTEM

The proposed sarcasm detection method utilizing machine learning techniques. The subsequent sections will provide a detailed explanation of each module, elucidating its role and functioning in the process.

A. Analyzing Data

Data analysis is an indispensable process that involves thorough examination, cleansing, transformation, and modeling of data to unearth valuable insights, draw meaningful conclusions, and facilitate well-informed decision-making. For this particular project, a dataset comprising 76,799 sarcastic messages was collected from Kaggle's Customer Support Machine Learning Group (ULB). Upon obtaining the data, meticulous processing and organization were carried out to address issues like incompleteness, duplicates, and errors. Subsequently, the cleaned dataset underwent various analytical techniques, including mathematical formulas like correlation and regression, to identify relationships among variables. This analytical approach empowers researchers to gain deeper insights into underlying patterns and trends within the data, thereby extracting valuable information and knowledge for future use.

B. Feature Selection

Feature selection is a critical aspect of data analysis, involving the careful identification of relevant features to be extracted from the data while discarding unnecessary ones. In the context of Twitter, users often employ both emotions and words to express their thoughts, making emotions a significant feature for detecting sarcastic sentences. In this project, we have introduced 21 distinct features in addition to standard unigrams and bigrams for the classification process. These 21 features are further categorized into four groups: Text expression-based features, Emotion-based features, Familiarity-based features, and Contrast-based features. By incorporating this diverse set of features, we aim to enhance the accuracy of our sarcasm detection model and ensure its effectiveness in identifying sarcastic statements.

C. Preprocessing

Social media data poses a challenge as it is unstructured, requiring thorough cleaning before training a sentiment analysis model. The accuracy of the results heavily relies on the quality of data used. Preprocessing a Twitter dataset involves several essential tasks, such as eliminating irrelevant information like special characters, emojis, and excessive blank spaces. Additionally, the data is formatted for better readability, and duplicates or very short tweets are removed. By performing these preprocessing steps diligently, the data becomes refined and ready for training the sentiment analysis model, leading to more accurate and reliable outcomes.

2.2.1 Feature Extraction and Feature Engineering:

At the outset of text analysis, the process of tokenization is applied to segment the text into individual words or tokens. Subsequently, these tokens are assigned numerical values, either in the form of integers or floating-point representations, which serve as input for the machine learning algorithm during the feature extraction phase. This crucial step holds immense significance in the system's development.

In the feature extraction process, the words undergo further treatment, involving reduction and stemming, which brings them to their base or root form [7]. This normalization process ensures that different variations of the same word are considered as a single entity, enhancing the efficiency of analysis and pattern recognition.

Feature extraction is a fundamental part of the workflow as it transforms raw text data into a format that machine learning algorithms can effectively comprehend. By representing words numerically, the algorithm can discern patterns, relationships, and trends within the data, leading to valuable insights and achieving heightened accuracy in predictions.

Feature cleaning

After the feature extraction we'll search for any null parameters in the data. If there is any null parameters there means we need to fill the parameters with the related content. In this process the steaming of the words are done. The similar words have been considered as one and the model is being built.

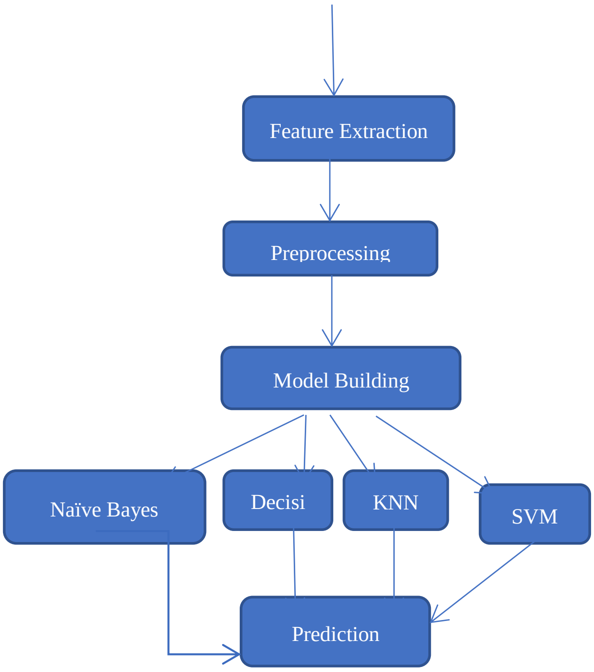

Fig 2.1 : Architecture of proposed System

D. MODEL BUILDING

The model is built according to the features. If the feature has labels the model can be built as supervised algorithm if the feature doesn't have the labels we need to use unsupervised learning algorithms if there is combination means Semi supervised [2] ensemble model should be used.

E. NAÏVE BAYES

Naïve Bayes [1] is based on two assumptions. First, all features in an entrance that needs to be classified are causative evenly in the decision (equally important). Secondly, all attributes are statistically self-determining, meaning that, knowing an attribute's value does not indicate whatever thing about other attributes' value which is not always true in practice. The process of classifying an instance is done by applying the Bayes rule for each class given the occurrence

F. DECISION TREE

There are two categories of decision trees: Classification trees & Regression trees. The decision tree is the creation of a tree from training tuples. A decision tree consists of nodes that form a tree structure; the topmost nodes are called the root node. Each non-leaf node denotes a test on a characteristic; each division represents the result of a test, and each leaf node hold a class label. Leaf nodes represent classes that are return it reaches the final node of the tree. The model is able to classify the instance by traverse the decision tree.

G. K- NEAREST NEIGHBORS

The k-Nearest Neighbors (kNN) algorithm is a simple occurrence based algorithm that plots all training instances and classify unlabeled instances based on their nearby neighbors. In instance based learners instance themselves are used to characterize the model not like the decision tree algorithms that use instance to develop a tree and that tree represent the model. However, it is argued that all learning algorithms are occurrence based while they all use instances of the training set to make models. In the kNN technique, the nearest neighbors are calculated by using any of the distance metric.

H. SUPPORT VECTOR MACHINE

SVM used to solve binary classification problems are comprehensive to nonlinear regression problems. SVMs are based on structural risk minimization dissimilar ANNs which is based on experiential risk minimization. The input data are transferred into a multidimensional feature break. After this alteration the SVM finds the best overexcited plane within the feature space.

I. Prediction

Metric Score

Metric score is the model constructed can't implement in real world without testing. So we'll test the model using test set and we are having the original values. After the model's prediction it is compared with the original values and we can find the metric score called Accuracy.

Real Time Detection:

After calculating metric score if the metric score is less means we can't use in real time so change the model and if the accuracy is good means we can implement in real world.

3. RESULTS AND DISCUSSION

First find the ROC curve and we can determine whether our ROC curve is good or not by looking at AUC (Area Under the Curve) and other parameters which are also called as Confusion Metrics. A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. All the measures except AUC can be calculated by using left most four parameters.

Table 3.1 : Predicted Class for ROC

		Predicated Class	
		Class=yes	Class=no
Actual Class	Class=yes	True positive	False Negative
	Class=no	False positive	True negative

Once calculated these four parameters then we can calculate Accuracy, Precision, Recall and F1 score.

Accuracy - Accuracy is that the most intuitive performance live and it's merely a magnitude relation of properly foretold observation to the entire observations. Therefore, you have got to appear at different parameters to judge the performance of your model. For our model, we've got 0.803 which means our model is approx. 80% accurate.

Accuracy = TP+TN/TP+FP+FN+TN

Precision - Precision is the magnitude relation of properly foretold positive observations to the entire foretold positive observations. The question is that this metric answer is of all passengers that labeled as survived, what percentage really survived? High preciseness relates to the low false positive rate. We have got 0.788 preciseness that is pretty sensible.

Precision = TP/TP+FP

Recall (Sensitivity) - Recall is that the magnitude relation of properly foretold positive observations to the all observations in actual category - affirmative. The question recall answers are: Of all the passengers that actually survived, what percentage did we tend to label? We have got recall of 0.631 that is nice for this model as it's on top of 0.5.

Recall = TP/TP+FN

F1score - F1 Score is that the weighted average of precision and Recall. Therefore, this score takes each false positives and false negatives under consideration. Intuitively it is not as straightforward to know as accuracy, however F1 is sometimes a lot of helpful than accuracy, particularly if you have got AN uneven category distribution. Accuracy works best if false positives and false negatives have similar price. If the value of false positives and false negatives area unit terribly totally different, it's higher to appear at each preciseness and Recall. In our case, F1 score is 0.701.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

4. CONCLUSION

The extracted features are successfully analyzed as sarcastic sentences by using various machine learning approaches such as naïve bayes, decision tree, kNN and SVM. Then the accuracy of various models can be classified by using classifiers for getting better accuracy.

References

1] MuhammedAbuliash, Ashraf , "Self deprecating sarcasm detection: An Amalgamation of Rule-Based and Machine Learning Approach", IEEE/WIC/ACM International Conference on Web Intelligence,2018.

[2] MondherBouazizi, TomoakiOtsuki, "A Pattern-Based Approach for Sarcasm Detection on Twitter", IEEE Digital Object Identifier 2016.

[3] ShaliniRaghav, Ela Kumar, "Review of Automatic Sarcasm Detection", 2nd International Conference on Telecommunication and Networks,2017.

[4] Edwin Lunando, AyuPurwarianti, "Indonesian Social Media Sentiment Analysis WithSaracsm Detection", ICASIS 2013.

[5] Manoj Y Manohar, Prof. PallaviKulkarni, "Improvement Sarcasm Analysis using NLP and Corpus based Approach", International Conference on Intelligent Computing and Control Systems 2017.

[6] ShubhamRendalkar, ChaitaliChandankhede, "Sarcasm Detection of Online Comments Using Emotion Detection, ICRICA 2018.

[7] Satoshi hiai, Kazutaka Shimada, "A Sarcasm Extraction Method Based on Patterns of Evaluation Expressions", 5^{th} IIAI International Congress on Advanced Applied Informatics,2016.