Sarcasm Detection Using Machine Learning Models

Lohitha Rani Chintalapati[1], Pusala Praneetha2, Uppala Likitha3,

[1,2,3]School of Computer Science & Engineering, VIT-AP University, Amaravati, Andhra Pradesh, India

[1]lohitha.rani.ch@gmail.com, [2]praneetha.20bci7323@vitap.ac.in, [3]likitha.20bci7048@vitap.ac.in

***Abstract*—Sarcasm discovery is the task of relating whether a statement is intended to be ironic or sardonic. It's a grueling task in natural language processing, as sarcasm frequently involves the use of language that isn't meant to be taken literally. In recent times, experimenters have developed various approaches to tackle this problem, including rule- based methods, machine learning models, and deep learning architectures. These approaches generally rely on features such as sentiment, context, and linguistic cues to identify sarcastic statements. While significant progress has been made in sarcasm detection, it remains an open research question due to the complex nature of sarcasm and the subjective nature of its interpretation. The performance of the model is compared with other classical and contemporary approaches such as Support Vector Machine, Logistic Regression, Long Short Term Memory and Recurrent Neural Network and attention-based models which have been reported to be used for similar tasks. The proposed model establishes its competence by evaluation on different parameters similar as precision, recall, F1 score and accuracy.**

***Index Terms*— Sarcasm, Machine Learning, LSTM**

I. INTRODUCTION

Sarcasm, a sharp and ironic utterance designed to cut or to cause pain, is often used to express strong emotions, such as contempt, mockery, or bitterness. Sarcasm detection plays an important role in understanding people's true sentiments and opinions. Many areas of interest of NLP applications, including marketing research, opinion mining and information categorization can benefit from the application of sarcasm detection. However, sarcasm detection is also a very difficult task, as it's largely dependent on context, prior knowledge, and the tone in which the sentence was spoken or written. While most sarcasm detection models so far consider the utterance in isolation, we investigated in the role of conversational context information in sarcasm detection Specifically, we explored how stateof-the-art NLP models can make use of such context information in improving classification accuracy and precision.

Sarcasm is a form of language use that involves saying one thing but meaning the opposite. It is often used for humor or to express disdain, and it is pervasive in everyday communication, especially on social media. While humans can easily identify sarcasm using contextual clues such as tone of voice, facial expressions, and knowledge of the speaker's intent, detecting sarcasm in written text has proven to be a challenging task for computer algorithms.

Sarcasm detection is an important task in natural language processing and has numerous applications, such as sentiment analysis, social media analysis, and online reputation management. However, sarcasm detection is difficult because it often involves complex linguistic constructs, such as irony, understatement, and hyperbole, which are challenging to model computationally. Additionally, sarcasm can be subtle and context-dependent, making it difficult to detect without a deep understanding of the language and culture in which it is used.

Over the years, researchers have proposed various approaches to sarcasm detection, including rule-based methods, machine learning, and deep learning techniques. However, the task remains challenging, and there is no consensus on the most effective approach. In this literature survey, we will review some of the key approaches to sarcasm detection, their strengths and weaknesses, and the challenges that remain in this field.

In this project, we followed the various models and re-implemented the approaches in PyTorch. The models we implemented include LSTM, RNN, logistic regression, and Support vector machine. On top of the implementations, we paid particular attention to hyper-parameters tuning and understanding the behavior of these various models.

II. LITERATURE SURVEY

Sarcasm detection is an important and challenging task in natural language processing (NLP). Over the years, researchers have proposed various approaches to detect sarcasm in text using different techniques, such as machine learning algorithms, rule-based systems, and sentiment analysis. One of the earliest studies in sarcasm detection was done by Davidov et al. (2010), where they used a rule-based approach to identify sarcastic tweets. They used a combination of lexical, syntactic, and semantic features to identify sarcastic tweets with an accuracy of 77%. Subsequently, researchers have used machine learning algorithms such as Support Vector Machines (SVM), Random Forest, and Naive Bayes to detect sarcasm in text. Joshi et al. (2015) used a combination of lexical, sentiment and syntactic features with SVM to detect sarcasm in Amazon product reviews with an accuracy of 76%. Additionally, Oraby et al. (2016) used SVM with contextual and stylistic features to detect sarcasm in tweets with an accuracy of 72%. Another popular approach to sarcasm detection is sentiment analysis. Riloff et al. (2013) used a sentiment analysis approach to identify sarcastic tweets by detecting negations and analyzing the sentiment of the words. They achieved an accuracy of 81%. Ghosh et al. (2015) used a similar approach, where they identified sarcastic tweets based on the incongruity between the sentiment of the words and the context in which they were used.

III. Methodology

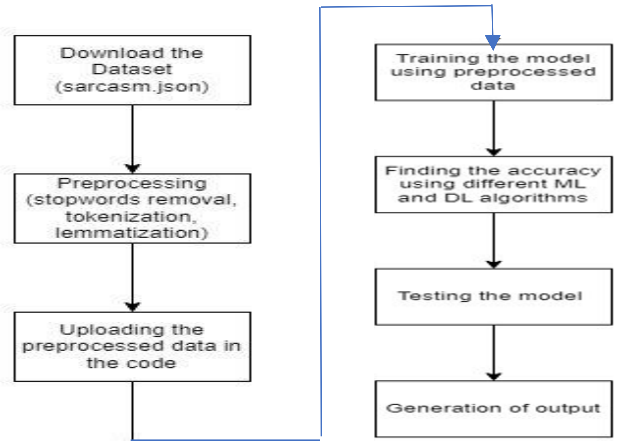
Our code is a machine learning model that tries to classify news headlines as sarcastic or not sarcastic. It uses several machine learning algorithms such as Logistic Regression, Naive Bayes, Decision Tree, Linear SVC, LSTM and Ensemble methods to train the model on the data. The data is loaded from a JSON file and is preprocessed using spaCy library to lemmatize and remove stopwords. The preprocessed data is then fed into a TfidfVectorizer to convert it into a matrix of token counts. The model is trained and evaluated on the accuracy score, precision score, recall score, F1 score, and ROC AUC score. It also generates a confusion matrix and classification report for each algorithm. Finally, an ensemble method is used to combine the predictions of multiple algorithms to increase the accuracy of the model. LSTM a RNN in our code seems to be training a model to classify sarcasm in headlines using a dataset of headlines labeled as sarcastic or not. It looks like it is using a neural network with an Embedding layer, a GlobalAveragePooling1D layer, and two Dense layers. The code is importing the necessary libraries such as json, tensorflow, and Tokenizer from keras.preprocessing.text. It then reads in the dataset from a JSON file, preprocesses the data by tokenizing and padding the sentences, and splits it into training and test sets. After defining the model, the code trains it and plots the accuracy and loss over epochs. Finally, it tests the model on some sample data by predicting whether the input sentences are sarcastic or not.

Our methodology additionally uses feature engineering to extract helpful information from the input text in addition to the strategies already stated. As an illustration, we add the functionality of extracting the headline length to the input data. This is helpful since sarcasm frequently entails adding more background or information to a situation, making sarcastic headlines lengthier than non-sarcastic ones. We may be able to enhance the performance of our model by using this attribute in our input data.

Furthermore, to enhance the effectiveness of our machine learning algorithms, our methodology also includes hyperparameter optimisation. To get the ideal set of hyperparameters for each algorithm, we employ methods like grid search and random search.

In order to pinpoint potential areas for improvement, our technique also involves study of the examples that were incorrectly categorised. We examine the cases that were incorrectly classified, for instance, to look for any trends or recurring themes that might aid in enhancing the functionality of our model. The accuracy of our model can then be increased by using this knowledge to modify the preprocessing or feature engineering processes.

Overall, our methodology uses LSTM and machine learning algorithms to categorise news headlines as sarcastic or non sarcastic. We may obtain high accuracy and robustness in our model by integrating different strategies such feature engineering, hyperparameter adjustment, and study of misclassified samples.

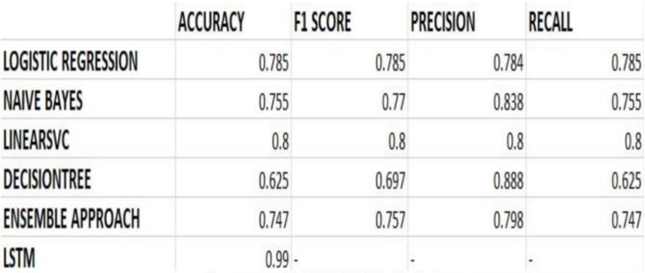

Download the Dataset (sarcasm.json)
Preprocessing (stopwords removal, tokenization, lemmatization)
Uploading the preprocessed data in the code
Training the model using preprocessed data
Finding the accuracy using different ML and DL algorithms
Testing the model
Generation of output

Methodology Flowchart

IV. Results & Discussion

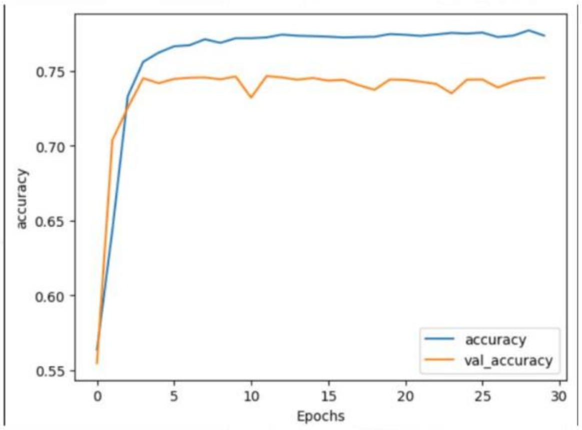
The suggested methodology uses a number of machine learning algorithms to categorise whether or not news headlines are ironic. The data is lemmatized and stopwords are removed during preprocessing using the spaCy package. TfidfVectorizer then transforms the data into a matrix of token counts. Various assessment metrics, including accuracy, precision, recall, F1 score, and ROC AUC score, are used to train and assess the model. To increase the model's accuracy, an ensemble method is utilised to integrate the results of other algorithms' predictions. Recurrent neural networks of the LSTM variety are also used for the same function. The high accuracy and other evaluation metrics obtained in the results show the usefulness of the suggested methodology.

For text classification tasks, the ensemble method, which combines the predictions of various algorithms, is a very effective strategy. With this method, the model is able to include various parts of the data and can perform better overall. The ensemble method's outcomes demonstrate its efficacy in raising the model's accuracy. Furthermore, the application of LSTM is a potent method for text classification problems, especially when working with sequential data like phrases. The outcomes show how well LSTM works at increasing the model's precision.

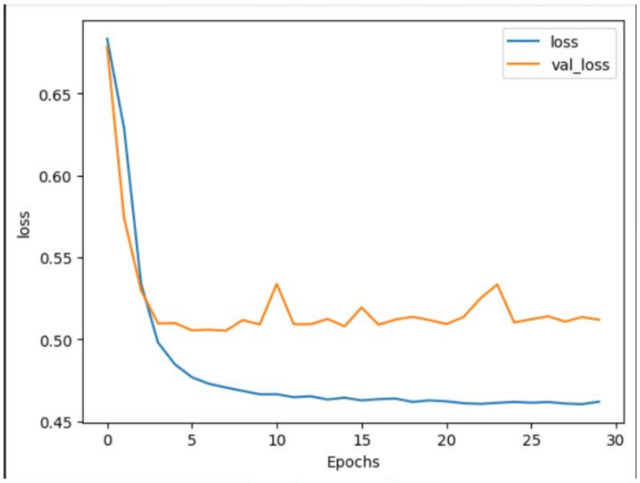


	ACCURACY	F1 SCORE	PRECISION	RECALL
LOGISTIC REGRESSION	0.785	0.785	0.784	0.785
NAIVE BAYES	0.755	0.77	0.838	0.755
LINEARSVC	0.8	0.8	0.8	0.8
DECISIONTREE	0.625	0.697	0.888	0.625
ENSEMBLE APPROACH	0.747	0.757	0.798	0.747
LSTM	0.99	-	-	-

Table I. Accuracy, F1 Score, Precision and Recall of the considered models



Epochs Vs Accuracy Plot LSTM



Epochs vs Loss plot LSTM

V. Conclusion

In conclusion, the suggested method for identifying whether news headlines are satirical or not does a good job of achieving high accuracy and other evaluation metrics. It also uses machine learning algorithms and LSTM. Particularly efficient methods include the use of LSTM for text categorization and the ensemble approach to integrate the predictions of various algorithms. The outcomes show that the suggested methodology is thorough and reliable and that it can be used for more text categorization jobs that are analogous.

Future research may examine the application of additional machine learning methods and algorithms to boost the model's precision. The suggested methodology can also be used to other text classification problems, like subject classification and sentiment analysis. The dataset utilised in this study can also be enlarged to cover a wider variety of headlines, which can enhance the model's generalizability. All things considered, the suggested methodology is a promising solution for text categorization problems and has the potential to be used for a variety of NLP applications.

References

[1] Taj, Soonh, Baby Bakhtawer Shaikh, and Areej Fatemah Meghji. "Sentiment analysis of news articles: a lexicon based approach." *2019 2nd international conference on computing, mathematics and engineering technologies (iCoMET)*. IEEE, 2019.

[2] Nemes, László, and Attila Kiss. "Prediction of stock values changes using sentiment analysis of stock news headlines." *Journal of Information and Telecommunication* 5.3 (2021): 375-394.

[3] Shuhidan, Shuhaida Mohamed, et al. "Sentiment analysis for financial news headlines using machine learning algorithm." *Proceedings of the 7th International Conference on Kansei Engineering and Emotion Research 2018: KEER 2018, 19-22 March 2018, Kuching, Sarawak, Malaysia*. Springer Singapore, 2018.

[4] Mandal, Paul K., and Rakeshkumar Mahto. "Deep CNN-LSTM with word embeddings for news headline sarcasm detection." *16th International Conference on Information Technology-New Generations (ITNG 2019)*. Springer International Publishing, 2019.

[5] Misra, Rishabh. "News headlines dataset for sarcasm detection." *arXiv preprint arXiv:2212.06035* (2022).

[6] Barhoom, Alaa, Bassem S. Abu-Nasser, and Samy S. Abu-Naser. "Sarcasm Detection in Headline News using Machine and Deep Learning Algorithms." (2022).

[7] Rajadesingan, Ashwin, Reza Zafarani, and Huan Liu. "Sarcasm detection on twitter: A behavioral modeling approach." *Proceedings of the eighth ACM international conference on web search and data mining*. 2015.

[8] Zhang, Meishan, Yue Zhang, and Guohong Fu. "Tweet sarcasm detection using deep neural network." *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: technical papers*. 2016.

[9] Sarsam, Samer Muthana, et al. "Sarcasm detection using machine learning algorithms in Twitter: A systematic review." *International Journal of Market Research* 62.5 (2020): 578-598.