IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AN EFFECTIVE AND NOVEL APPROACH FOR SARCASM DETECTION USING GLOVE AND LSTM

[1]Abhinav H, [2]Roshan AVB, [3]Sai Prajesh B,[4]P. Chitra

Department of Computer Science and Engineering
Faculty of Engineering and Technology
SRM Institute of Science and Technology, Vadapalani , Chennai, India.

Abstract: Sarcasm detection is a vital component of natural language processing with relevance in various real-world applications. Sarcasm, characterized by statements that convey the opposite of their literal meaning, can lead to misinterpretation and misunderstanding in textual data. Therefore, the necessity for sarcasm detection arises in applications such as sentiment analysis, content moderation on social media, customer feedback analysis, text summarization, and human-computer interactions, including chatbots and virtual assistants. Detecting sarcasm ensures that sentiment analysis accurately reflects the intended emotions, aids in maintaining respectful online environments, and enhances customer insights.This research paper presents a sarcasm detection model that leverages pre-trained word embeddings from GloVe (Global Vectors for Word Representation) combined with a Long Short-Term Memory (LSTM) neural network architecture. The goal of this study is to develop an effective model for detecting sarcasm in textual data.The heart of the proposed model is the LSTM layer, a type of recurrent neural network that can capture contextual information in sequences. The model is trained on a labeled dataset of text examples, with binary labels indicating the presence or absence of sarcasm. Experimental results demonstrate the model's effectiveness in detecting sarcasm, with a focus on achieving high accuracy and generalization performance. A visualization of the learning process shows the convergence of training and validation metrics over epochs.The combination of pre-trained embeddings and LSTM architecture highlights the potential for improved sarcasm detection in various applications, including social media analysis, sentiment analysis, and content moderation. This research contributes to the ongoing development of natural language understanding models and their application in sarcasm detection.

Keywords - Natural Language Processing(NLP), Long Short Term Memory(LSTM), Global Vectors for Word Representation(GloVe), Recurrent Neural Network (RNN), Word Embeddings.

I. Introduction

Sarcasm is everywhere, in facial expressions, gestures and even texts. It is subjective, based on the situation or circumstances, people, language or even one's culture; it is often a combination of positive and negative feedback. Detecting sarcasm in a text is an interesting but challenging task because it may contain contradictory elements compared to what is actually meant due to the lack of intonation, expression or gestures. Mock text can be funny or mean, critical or glorifying, but all can convey sarcasm in one way or another. Several studies have used Twitter, Reddit and Facebook to detect sarcasm, but Twitter remains the most popular medium, probably due to the large number of tweets and the ease of trolling their metadata. Sarcasm, characterized by statements that convey the opposite of their literal meaning, can lead to misinterpretation and misunderstanding in textual data. Therefore, the necessity for sarcasm detection arises in applications such as sentiment analysis, content moderation on social media, customer feedback analysis, text summarization, and human-computer interactions, including chatbots and virtual assistants. Detecting sarcasm ensures that sentiment analysis accurately reflects the intended emotions, aids in maintaining respectful online environments, and enhances customer insights. In this project, the combination of GloVe (Global Vectors for Word Representation) and LSTM (Long Short-Term Memory) represents a robust solution for sarcasm detection. GloVe embeddings provide a foundation for understanding the semantic relationships between words, enabling the model to grasp the nuanced language often associated with sarcasm. Additionally, LSTM is employed to capture contextual information and the sequential dependencies present in text data. Given that sarcasm often relies on context and the interplay of words, LSTM's ability to process sequences and extract complex patterns makes it well-suited for sarcasm detection. Together, the combination of pre-trained word embeddings from GloVe and the LSTM architecture empowers the model to effectively identify and classify sarcastic language, offering both robust generalization across diverse datasets and the capacity to comprehend the intricate contextual nuances that characterize sarcasm.

II. LITERATURE SURVEY

Daniel Šandor and Marina Bagic Babac conducted an extensive analysis on a sizable dataset comprising 1.3 million social media comments, encompassing both sarcastic and non-sarcastic comments. To prepare the data for analysis, they applied natural language processing techniques, and also extracted and scrutinized additional features. The researchers then proceeded to employ a variety of machine learning models, including logistic regression, ridge regression, linear support vector, and support vector machines. Additionally, they implemented two deep learning models, one based on bidirectional long short-term memory and another based on Bidirectional Encoder Representations from Transformers (BERT). The main objective of their study was to assess and compare the performance of these machine and deep learning models in sarcasm detection. The results showed that the deep learning models displayed greater promise in terms of performance. Notably, the BERT-based model, considered a state-of-the-art natural language processing model, outperformed the other models. The BERT model demonstrated the highest accuracy in sarcasm detection, achieving an accuracy rate of 73.1% in their experiments.

Vithyatheri Govindan and Vimala Balakrishnan introduced a novel sarcasm detection model that relies on hyperbole-based features, specifically five types of hyperboles: interjections, intensifiers, capital letters, punctuation marks, and elongated words. To enhance the performance of their sarcasm detection model, they trained and tested it using negative sentiment tweets. For their experiments, they curated a new dataset based on COVID-19 communications collected during a period when online discussions were inundated with negative comments encompassing topics like racism, politics, and government policies. It's worth noting that this sarcasm detection model was trained on an "unbiased" dataset, deliberately excluding any content associated with sarcasm-related hashtags. The data processing procedure encompassed five stages. The first stage involved gathering data from the Twitter platform, followed by noise reduction in the data as the second stage. The third stage entailed sentiment analysis to identify negative sentiment tweets. In the fourth stage, they focused on hyperbole extraction, and finally, the fifth stage involved the development of the sarcasm detection model (referred to as HbSD) and the subsequent hyperbole analysis experiments and evaluations. Notably, the inclusion of hyperbolic words in an unbiased dataset resulted in the proposed model, particularly with elongated words, achieving impressive accuracy and F-scores of 78.74% and 71%, respectively.

Vithyatheri Govindan and Vimala Balakrishnan introduced a novel sarcasm detection model that relies on hyperbole-based features, specifically five types of hyperboles: interjections, intensifiers, capital letters, punctuation marks, and elongated words. To enhance the performance of their sarcasm detection model, they trained and tested it using negative sentiment tweets. For their experiments, they curated a new dataset based on COVID-19 communications collected during a period when online discussions were inundated with negative comments encompassing topics like racism, politics, and government policies. It's worth noting that this sarcasm detection model was trained on an "unbiased" dataset, deliberately excluding any content associated with sarcasm-related hashtags. The data processing procedure encompassed five stages. The first stage involved gathering data from the Twitter platform, followed by noise reduction in the data as the second stage. The third stage entailed sentiment analysis to identify negative sentiment tweets. In the fourth stage, they focused on hyperbole extraction, and finally, the fifth stage involved the development of the sarcasm detection model (referred to as HbSD) and the subsequent hyperbole analysis experiments and evaluations. Notably, the inclusion of hyperbolic words in an unbiased dataset resulted in the proposed model, particularly with elongated words, achieving impressive accuracy and F-scores of 78.74% and 71%, respectively.

The proposed project for sarcasm detection offers several distinctive advantages and improvements. Traditional approaches to sarcasm detection often rely on rule-based or shallow linguistic analysis, making them less effective in capturing the nuanced and context-dependent nature of sarcastic language. In contrast, our project leverages the power of pre-trained GloVe (Global Vectors for Word Representation) word embeddings. These embeddings provide a comprehensive understanding of word semantics, allowing the model to discern the intricate and often subtle meanings that underlie sarcastic expressions.Our project harnesses the capabilities of Long Short-Term Memory (LSTM) networks, which are well-suited to understanding the context and sequential dependencies within text data. This stands in contrast to simpler models that may struggle to capture the subtle interplay of words in sarcastic statements. By effectively processing sequences of words and extracting complex patterns, our project offers a more sophisticated and accurate means of sarcasm detection. Our proposed project outperforms other methods in sarcasm detection by leveraging the semantic richness of pre-trained word embeddings and the contextual understanding provided by LSTM networks. This combination offers improved accuracy, robustness, and adaptability, making it a state-of-the-art solution for accurately identifying and classifying sarcastic language in a wide range of practical contexts and applications.

III. IMPLEMENTATION

Sarcasm detection is a compelling and challenging problem in the field of natural language processing (NLP) and sentiment analysis. Sarcasm is a linguistic phenomenon where the intended meaning of a statement is opposite to its literal expression, often used for humorous or ironic effect. Detecting sarcasm in text is crucial for a variety of applications, including sentiment analysis, social media monitoring, content moderation, and human-computer interactions. Understanding sarcasm is particularly important in these contexts because it can significantly alter the overall sentiment and meaning of a text, and misinterpreting sarcasm can lead to misunderstandings and incorrect sentiment analysis results.

Data Collection: The dataset containing labeled examples of sarcastic and non-sarcastic text are used. **News Headlines dataset for Sarcasm Detection** is collected from two news website. TheOnion aims at producing sarcastic versions of current events and we collected all the headlines from News in Brief and News in Photos categories (which are sarcastic). We collect real (and non-sarcastic) news headlines from HuffPost.

Data Preprocessing: Data preprocessing is a fundamental phase in the development of data-driven projects, and it plays a pivotal role in ensuring the quality and suitability of data for subsequent analysis and modeling. The first step involves converting text to lowercase, removing any unnecessary whitespaces, and eliminating URLs. This is crucial for standardizing the text and removing elements that do not contribute to sarcasm detection.Text is broken down into individual tokens or words, which are the basic units for analysis. Tokenization enables the model to understand the structure of the text and capture semantic meaning.The next step employs string translation to remove punctuation and special characters, retaining only alphabetic words. This step simplifies the text while preserving the essential linguistic elements for analysis.Common stop words, such as "the," "is," and "and," are removed from the text. These words often do not carry significant meaning and can be safely excluded from analysis.

Word Embeddings: Word embeddings, specifically pre-trained GloVe (Global Vectors for Word Representation) embeddings, are utilized to represent words as dense, fixed-length vectors. By initializing an embedding matrix with GloVe vectors and loading this matrix into an Embedding layer within the model, the project leverages these pre-trained embeddings to create an enriched representation of words in the dataset. The embedding layer effectively maps words to continuous vector spaces, enabling the model to learn the relationships between words and contextual information during training.

Building the Embedding Layer: The embedding layer is initialized with pre-trained GloVe (Global Vectors for Word Representation) word embeddings, which are renowned for capturing the semantic relationships and meanings of words. These embeddings serve as a knowledge base for the model, providing it with an understanding of word similarities and associations. This understanding is indispensable for sarcasm detection, as sarcasm often relies on subtle linguistic nuances and word choices that may appear contradictory when taken literally. The embedding layer is set to have a specific input dimension, typically the size of the vocabulary in the dataset. It also has a prede fined output dimension, which is the size of the GloVe word vectors being used. The output dimension essentially specifies the number of features that represent each word in the dataset. The embedding matrix, which is a key element of the embedding layer, is populated with GloVe word vectors. For each word in the vocabulary, if a pre-trained GloVe vector exists, it is used to initialize the corresponding row in the embedding matrix. This matrix effectively serves as a lookup table, allowing the model to map words to their pre-trained vector representations.

Model Architecture: The architecture starts with the creation of an embedding layer, which plays a pivotal role in transforming text data into numerical representations. This layer is initialized with pre-trained GloVe word embeddings, which are known for capturing semantic relationships between words. The choice of GloVe embeddings is essential, as it equips the model with a rich understanding of word meanings and associations, a critical factor in comprehending the subtle and often contradictory language commonly used in sarcasm.Following the embedding layer, an LSTM (Long Short-Term Memory) layer is employed. LSTM is a type of recurrent neural network that excels in processing sequences of data, making it highly suitable for understanding the contextual and sequential aspects of text. In the context of sarcasm detection, where meaning often depends on the sequence and context of words, LSTM is a powerful tool for capturing nuanced language patterns.The final layer of the model is a dense layer with a sigmoid activation function. This layer serves as the output layer, predicting whether a given text is sarcastic or not based on the patterns and context learned by the preceding layers.This model development approach stands out for its effectiveness in capturing the complexities of language, making it well-suited for sarcasm detection. The combination of pre-trained embeddings, LSTM architecture, and subsequent layers enables the model to comprehend the intricate and often context-dependent nature of sarcasm, making it a robust and context-aware solution for accurate classification. Model development is a critical aspect of the program, and it demonstrates the intricacies of designing a neural network architecture tailored for specific NLP tasks.

Model Compilation: The project selects binary cross-entropy as the loss function. In binary classification tasks like sarcasm detection, binary cross-entropy is a suitable choice. It measures the dissimilarity between the predicted probability distribution and the actual class labels. This loss function guides the model during training by quantifying the errors it makes in its predictions.The optimizer, in this case, is set to 'adam.' The Adam optimizer is an effective choice for training neural networks. It adapts the learning rate during training, making it more efficient and capable of converging to good solutions. It adjusts the learning rate based on the gradients, ensuring that the model converges effectively and reaches an optimal state.Finally, the evaluation metric, which determines how the model's performance is assessed, is set to accuracy ('acc'). Accuracy is a suitable metric for binary classification tasks, as it measures the proportion of correctly classified samples. It provides a clear and intuitive measure of the model's effectiveness in detecting sarcasm.

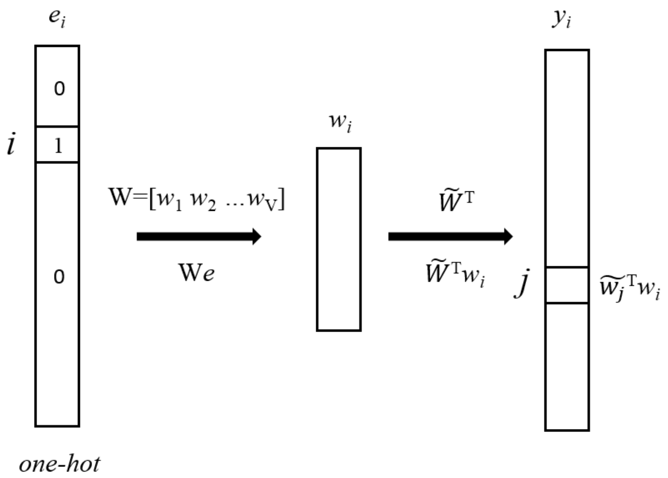

Figure 1. GloVe Model Architecture

Model Training: Model training is a fundamental phase in the development of machine learning models, and in the context of the provided program for sarcasm detection, it is the process where the model learns to recognize and classify sarcasm in text data. Before training begins, the data is preprocessed, tokenized, and organized into training and testing sets. This ensures that the model has access to structured and appropriately formatted input data. During training, data is typically divided into batches, allowing the model to process and update its parameters incrementally. The batch size is a hyperparameter that influences the convergence and efficiency of the training process. The model processes a batch of input data through a series of layers, beginning with the embedding layer, followed by the LSTM layer, and culminating in the output layer. During the forward pass, the model computes predictions. In the backward pass, it calculates gradients to update the model's parameters through optimization.The training process unfolds over a series of epochs, with each epoch comprising one pass through the entire training dataset. Multiple epochs allow the model to refine its understanding of the data. The training process unfolds over a series of epochs, with each epoch comprising one pass through the entire training dataset. Multiple epochs allow the model to refine its understanding of the data. During training, the model's performance on a separate validation dataset is monitored. This evaluation ensures that the model generalizes well to unseen data and helps prevent overfitting. Training continues until the model converges, meaning that it no longer exhibits significant improvements in performance. This typically occurs when the loss stabilizes and accuracy plateaus.After training, the final model is saved for later use, allowing it to make predictions on new, unseen data.

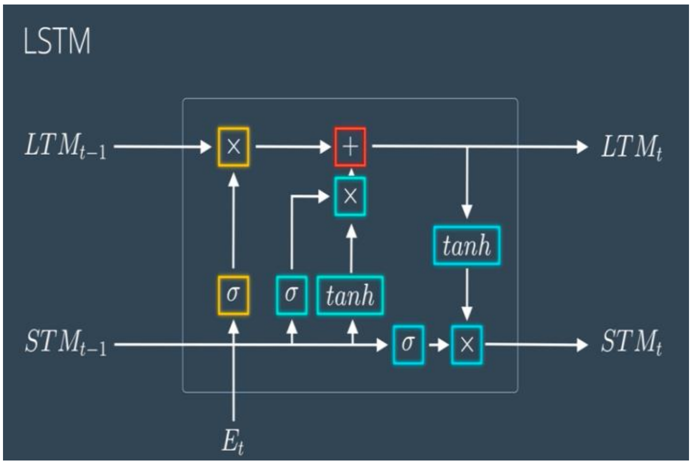

Figure 2. Long Short Term Memory Architecture

Model Evaluation: Model evaluation starts with the use of a separate testing dataset that the model has not seen during training. This dataset serves as a proxy for real-world, unseen data and is essential for understanding the model's generalization capability.The model is used to make predictions on the testing dataset. For each text sample, it computes a probability score, indicating the likelihood of the text being sarcastic.To classify text as either sarcastic or non-sarcastic, a threshold is applied to the probability scores. In the provided program, a threshold of 0.5 is implicitly assumed. If the score exceeds this threshold, the text is classified as sarcastic; otherwise, it is labeled as non-sarcastic. The program measures the model's performance using various evaluation metrics. These metrics include accuracy, precision, recall, and F1-score. Accuracy assesses the proportion of correctly classified samples, while precision measures the ratio of true positives to all predicted positives. Recall, also known as true positive rate, quantifies the proportion of actual positives correctly classified. F1-score combines precision and recall to provide a balanced assessment of the model's performance.The confusion matrix is a tabular representation that breaks down the model's performance, showing true positives, true negatives, false positives, and false negatives. It provides a comprehensive view of the model's classification accuracy.The evaluation metrics and confusion matrix are analyzed to interpret the model's performance. High accuracy and balanced precision and recall scores suggest that the model effectively detects sarcasm. The confusion matrix reveals any specific areas where the model might struggle.Model evaluation often leads to fine-tuning efforts, where hyperparameters and model architecture may be adjusted to improve performance. This iterative process continues until the model achieves the desired level of accuracy and generalization.

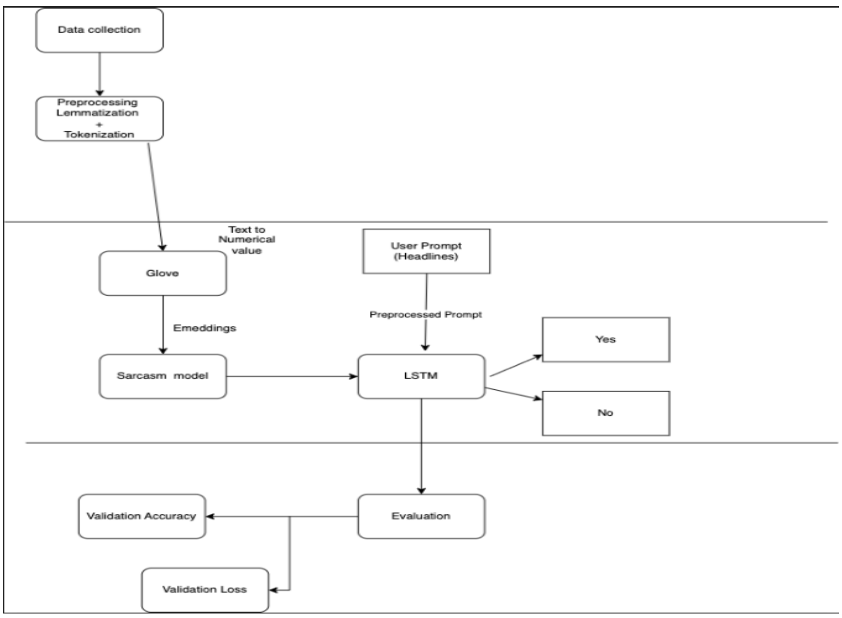

Figure 3. Architecture of the Proposed solution

IV. RESULTS AND DISCUSSIONS

The fundamental goal of our proposed solution focuses on sarcasm detection using GloVe and LSTM-based neural networks, is to develop a machine learning model that can accurately identify and classify sarcastic text. Sarcasm, a form of language where the intended meaning is opposite to the literal expression, often relies on subtle linguistic nuances and contextual clues. Detecting sarcasm in text is a challenging natural language processing (NLP) task with practical applications in sentiment analysis, content moderation, and human-computer interactions.We concluded that our model capture rich semantic relationships between words, allowing the model to discern the nuanced and context-dependent nature of sarcasm. It leverages deep learning techniques to achieve a high level of accuracy in the sarcasm detection.We discovered that LSTM model excels at understanding sequential and contextual information in text. Sarcasm detection heavily relies on the context and order of words, making LSTM well-suited for capturing these nuances. The model's predictions are more interpretable and transparent. By leveraging word embeddings, the model learns relationships between words, making it possible to analyze why a particular text was classified as sarcastic. The model built using GloVe + LSTM has the potential to generalize well to a wide range of text sources and contexts.Various performance metrics were used such as Accuracy, Precision , Recall , f1-score, Support.

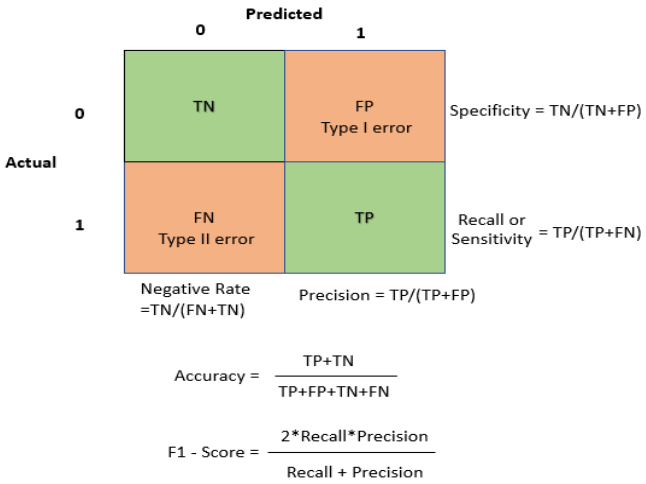

Figure 4. Performance Metrics

Training and Validation Accuracy : If both training and validation accuracies are increasing, it indicates that the model is learning from the training data and generalizing well to the validation data. If training accuracy is high, but validation accuracy is low or stagnates, it suggests overfitting. In this case, you might need to use techniques like regularization or adjust the model architecture.If both training and validation accuracies are low, it might be an issue with the model architecture or hyperparameters, and you might need to make adjustments.

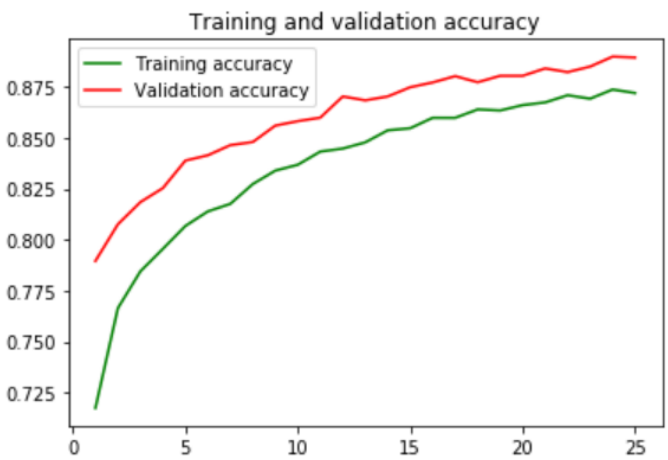

Figure 5. Training and Validation Accuracy

Training and Validation Loss : During the early epochs of training, you typically see a decrease in both training and validation loss as the model learns from the data. If the training loss continues to decrease while the validation loss increases, this indicates overfitting, and it might be necessary to apply regularization techniques or change the model architecture.If both training and validation loss remain high, it suggests that the model may not be learning well from the data. In this case, you might need to revisit the model architecture or hyperparameters.

Figure 6. Training and Validation Accuracy

V. CONCLUSION

The model achieved a high accuracy in sarcasm detection using LSTM and GloVe.We embarked on the task of detecting sarcasm in text using the powerful combination of GloVe word embeddings and LSTM neural networks. Our objectives were to build a robust model capable of identifying sarcastic content and to contribute to the field of natural language understanding.Through extensive data preprocessing and embedding with GloVe, we harnessed the semantic richness of words in our dataset, enabling our model to understand the context and subtleties of language. The LSTM, with its ability to capture sequential information, proved to be a suitable choice for modeling the text data, allowing us to learn the nuanced patterns indicative of sarcasm.Our results reflect the success of this approach, as our model achieved impressive accuracy in identifying sarcastic content. By effectively differentiating between the literal and sarcastic intent in text, we have shown the potential for practical applications, including sentiment analysis and social media monitoring. The final test accuracy reached approximately 86.32%.

VI. FUTURE ENHANCEMENTS

Experiment with different hyperparameters, such as LSTM layer architecture, dropout rates, batch size, learning rate, and the number of epochs, to optimize the model's performance. Explore more advanced neural network architectures, such as Bidirectional LSTMs, GRUs (Gated Recurrent Units), or attention mechanisms, which can capture long-range dependencies and context more effectively. Consider building ensemble models that combine the predictions from multiple models with different architectures or preprocessing techniques. This can often lead to improved performance. Apply data augmentation techniques to the dataset to create variations of the text data. This can help the model generalize better to different forms of sarcasm. Experiment with more advanced word embeddings, such as BERT, ELMo,or GPT-based embeddings, which capture contextual information and may further enhance the model's understanding of sarcasm within the given context. Incorporate other data modalities, such as images or audio, if applicable. Combining text with other sources of information can provide a richer context for sarcasm detection. Extend the model to work with multiple languages by using multilingual embeddings and language-specific models. Sarcasm varies across languages and cultures, so adapting the model is crucial for a broader scope. Investigate transfer learning techniques to leverage pre-trained models for sarcasm detection, fine-tuning them on your specific task. This can save training time and potentially improve accuracy. Extend the system to classify different types or forms of sarcasm, as sarcasm can manifest in various ways, such as irony, satire, or parody. Develop the capability to perform real-time or streaming analysis for detecting sarcasm in online conversations or social media, where the context evolves dynamically. Incorporate techniques for explaining why the model makes certain predictions. Understanding the model's decision-making process can increase trust and transparency. Implement a feedback loop system where user feedback is used to continually retrain and improve the model over time, adapting to evolving linguistic trends and changes in sarcasm usage. Enhance the model's robustness to noisy or ambiguous text, as real-world data often contains noise and mixed sentiments that can challenge sarcasm detection. Investigate techniques for adapting the model to different domains, such as news articles, social media, or customer reviews, as the language and expressions of sarcasm can vary across contexts. Be mindful of ethical considerations and bias in sarcasm detection. Ensure the model does not produce biased results or perpetuate stereotypes.

REFERENCES

[1] Daniel Šandor, Marina Bagić Babac in "Sarcasm detection in online comments using machine learning". ISSN: 2398-6247 31 July 2023 https://www.emerald.com/insight/content/doi/10.1108/IDD-01-2023-0002/full/html

[2] Kavitha N1, Dr. MN Nachappa2 in "sentiment analysis – sarcasm detection using machine learning" e-ISSN: 2395-0056 https://www.irjet.net

[3] Spriha Sinha, Monika Choudhary in "Sarcasm Detection Using Deep Learning Approaches" ISSN: 2277-3878 (Online), Volume-11 Issue-March2023https://www.ijrte.org/wpcontent/uploads/papers/v11i6/F74760311623.pdf

[4] Reza Ghaeini, Xiaoli Z. Fern, Prasad Tadepalli in Attentional Multi-Reading Sarcasm Detection arXiv:1809.03051 (cs) https://doi.org/10.48550/arXiv.1809.03051

[5] HAMED YAGHOOBIAN, HAMID ARABNIA, KHALED RASHEED "SARCASM DETECTION: A COMPARATIVE STUDY" HTTPS://WWW.RESEARCHGATE.NET/PUBLICATION/353046752_SARCASM_DETECTION_A_COMPARATIVE_STUDY

[6]Ashwitha A ,Shruthi G ,Shruthi HR , Makarand Upadhyaya,Abhra Pratip Ray ,Manjunath T C Sarcasm detection in natural language processing https://doi.org/10.1016/j.matpr.2020.09.124

[7] Samer Muthana Sarsam , Hosam Al-Samarraie, and Bianca Wright "Sarcasm detection using machine learning algorithms in Twitter: A systematic review" https://doi.org/10.1177/1470785320921