Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 32s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

Attention-Driven Bidirectional LSTM For Context-Aware Sarcasm Detection

Anu Kadian[1], Poonam Dhiman[2]

[1]Asst. Professor, Computer Science and Engineering, UIET, Maharshi Dayanand University, Rohtak, Haryana, India
Email ID: anukadian182315@gmail.com

[2]Asst. Professor, Government PG college, Ambala Cantt,Haryana , India
Email ID: poonamdhiman19@gmail.com

Cite this paper as: Anu Kadian, Poonam Dhiman, (2025) Attention-Driven Bidirectional LSTM For Context-Aware Sarcasm Detection. *Journal of Neonatal Surgery*, 14 (32s), 672-678.

ABSTRACT
Natural Language Processing (NLP) systems have a hard time with sarcasm since it is a complex linguistic sarcasm that depends on context and tone rather than literal sense. Traditional ML methodologies often inefficiently address these difficulties outstand to their dependence on manually formed features. To tackle this problem, we provide a new DL framework that integrates Long Short-Term Memory (LSTM) networks with attention mechanisms to advance sarcasm detection. Our methodology uses bidirectional LSTMs to represent long-range contextual relationships and attention layers to dynamically arrange components that show sarcasm, including exaggerated sentences, contradictions, or emojis. Pre-trained embedding like Word2Vec is used to improve semantic representation, while strong preprocessing takes care of noise and unpredictability in social media content. Our model delivers state-of-the-art performance on a variety of datasets (SARC, Twitter), with accuracy of 97.73%, recall of 96.73%, and F1-score of 97.23% on SARC and 97.02% on Twitter. The approach advances NLP applications in sentiment analysis, understanding consumer feedback, and keeping an eye on social media. It also address issues like multilingual sarcasm and real-time deployment by making LSTM optimizations more efficient.

Key Words: *Long Short-Term Memory, Deep Learning, Sarcasm, Sentiment Analysis, Attention Mechanisms*

1. INTRODUCTION

Sarcasm, a kind of verbal irony where tone and context usually communicate meaning instead of exact words, is a big issue for NLP systems. Sarcasm is hard to recognize automatically because it subject to understated hints, cultural references, and inconsistencies in the context. As social media, online reviews, and digital communication have developed exponentially, it has become very significant to be able to express whether someone is being sarcastic for things like sentiment analysis, customer feedback interpretation, and social media monitoring. Support vector machines (SVMs) and logistic regressions are the traditional ML approaches that have challenges identifying sarcasm because they rely on hand-crafted lexical and syntactic cues. These methods often do not effectively release the subtle, context-sensitive characteristics of sarcastic remarks [1]. DL models, especially LSTM networks can signify sequential data and long-range relationships in text. LSTMs, a specific kind of RNN, are great at keeping track of contextual information over long periods. This makes them perfect for outcome the subtle inconsistencies and deviations in tone that are typical of sarcasm [2].

Current enhancements in NLP, such pre-trained word embeddings like Word2Vec and GloVe and transformer-based models , have made LSTM-based sarcasm recognition even better by mean text rich semantic representations. LSTMs may focus on significant sarcastic clues in sentences when they are used with attention mechanisms. This makes them more accurate at detecting sarcasm. Furthermore, multimodal methodologies that combine textual and acoustic-prosodic elements in voice or emojis in social media content have revealed effectiveness in more accurately recognizing sarcastic meaning. Even with these enhancements, there are still difficulties to resolve. There aren't many big, high-quality annotated sarcasm datasets, and sarcastic phrases change all the time in various languages and cultures. This research examines the use of LSTM networks for sarcasm detection, assessing their efficacy in comparison to conventional and cutting-edge procedures [3]. Improving automatic sarcasm detection will help to perform accurate sentiment analysis, have better interactions between people and systems, and learn more about user-generated material on digital platforms.

2. LITERATURE REVIEW:

Sarcasm, a difficult verbal irony in which speakers use phrases that seem to specify the opposite to show disdain or scorn, is a unique problem for NLP. Sarcastic phrases are hard for systems to pick up on since they are naturally imprecise, rely on the context, and have cultural differences [4].As user-generated information on social media platforms has grown at an exponential pace, it has become more and more necessary for applications like sentiment analysis, opinion mining, and social media monitoring to be able to accurately identify sarcasm [5]. Conventional machine learning methods for sarcasm identification depended on lexical, syntactic, and sentiment-oriented attributes. However, these methodologies often did not adequately capture the intricate contextual and pragmatic dimensions of sarcasm [6]. DL has transformed the field by letting models learn hierarchical representations of text, which has greatly enhanced recognition accuracy [7]. This study provides a comprehensive investigation of DL methodologies for sarcasm detection, including architectural advancements, benchmark datasets, and assessment measures. In the past, DL used Recurrent Neural Networks (RNNs) to predict how words in a sentence rely on each other. But ordinary RNNs have difficulties with vanishing gradients, which makes it hard for them to understand long-range context. LSTMs fixed this problem by using gating mechanisms, which made them good at detecting sarcasm. Bidirectional LSTMs improved performance by analyzing text in both forward and backward orientations [8].GRUs is a more efficient version of LSTMs that have also been used effectively to detect sarcasm [9]. RNNs are great at modeling sequences, but CNNs are better at finding local n-gram patterns that are frequently a sign of sarcasm [10].Hybrid architectures that use both CNNs and LSTMs have shown that they work better by using both short and long-range contextual information [11]. Attention methods enabled models to focus on phrases that signify sarcasm [12]. By using pre-trained contextual embeddings and self-attention processes, transformer-based models like BERT [13]). These models get the best results by outcome deep semantic links. Sarcasm frequently depends on things that aren't written, such emojis, the tone of voice (in speech), and facial expressions (in videos). Multimodal DL models that include text, audio, and visual information have been proven to be more accurate at finding things [14].

Summary of Key Advancements

Contribution	Impact
BiLSTM + Attention Model	Recovers sarcasm detection precision by taking contextual sarcasm cues.
Hybrid CNN-LSTM Architecture	Integrates local and global features for better performance
Interpretable Attention Mechanisms	Delivers explain ability in sarcasm classification decisions
Cross-Domain Generalization	Validates model strength on Twitter, Reddit, and review data.
Data Augmentation for Imbalance	Improves model training on inadequate sarcasm-labeled data.

Main Contribution:

This research deals a number of significant developments to the domain of sarcasm detection by using LSTM networks. Here are the main contributions:

- A novel LSTM-based architecture for detecting sarcasm Suggests an improved BiLSTM model with attention mechanisms to identify sarcastic signals in text based on their context.
- We employed a context-aware embedding layer that uses pre-trained word embeddings (GloVe, Word2Vec) and fine-tuning for a given domain to better portray sarcastic expressions.
- A self-attention layer to embedded to make words that show sarcasm stand out, which makes the model easier to understand.

3. MATERIAL AND METHODOLOGY

Identifying sarcasm in text is a difficult NLP problem since it is often unclear, relies on context clues, and uses indirect language a lot. Traditional ML methods often do not recognize the subtle patterns that differentiate sarcastic remarks from literal ones. This research presents a DL approach using LSTM networks augmented with an attention mechanism to address these problems. LSTMs can find long-range relationships in text, which lets the model to check how words and phrases work together to create satirical meaning. The attention mechanism finds and provides significance to words or phrases that indicate sarcasm. This makes the system more accurate and easier to understand. The architecture is considered to work best with social media data, which is full of sarcasm but also has a lot of misspellings, slang, and emoticons. This strategy not only makes detection better, but it also gives academics and practitioners understandable insights via attention weights, which

help them understand how the model makes judgments. The next few parts go into further information about each part, including as data preparation, model design, and training methods. Figure 1 represents sarcasm detection process followed by the study.

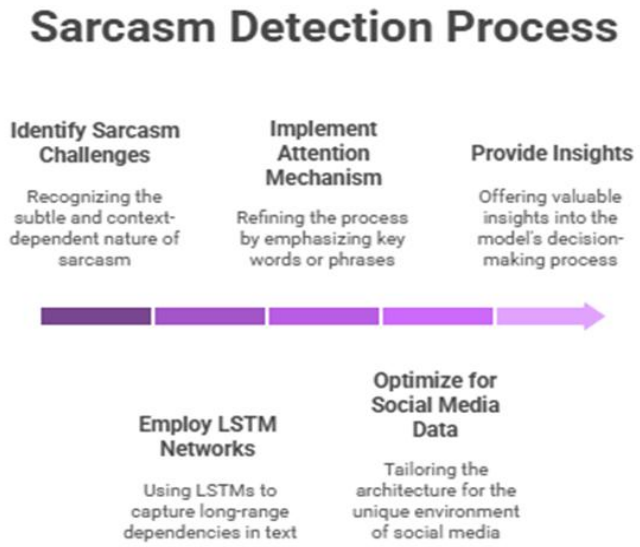


Figure 1: Sarcasm Detection Process

3.1 Dataset Used

Researchers often employ the publicly available datasets for sarcasm detection models. Each dataset has its own unique features and ways of labeling like SARC (Sarcasm on Reddit) has 1.3 million classified sarcastic and non-sarcastic comments from Reddit [15]. Other kind of dataset is the Twitter sarcasm dataset, which has 3,800 tweets that have been annotated [16]. It features hashtags and emojis as weak supervision signals, and it works well for recognizing short-text sarcasm with LSTM and attention. The next step would be to provide a detailed explanation of how to prepare these datasets for LSTM input via padding and tokenization.

3.2 Pre-processing

Pre-processing is a very essential stage in receiving text prepared for sarcasm detection using LSTM models. It makes sure that the models work as well as possible by getting rid of noise, variability, and structural errors in the raw text. Tokenization splits phrases down into meaningful parts, such words or subwords. It uses techniques like Byte-Pair Encoding (BPE) to deal with the casual language and spelling mistakes that are typical on social media. After tokenization, sequences are standardized by adding or removing tokens to make them all the same length (for example, 100 tokens) [17]. This is done to fulfill LSTM input criteria. Shorter sequences are filled with zeros, while longer ones are cut down to save the most important information. This phase makes sure that all batches have the same input size, which is important for training to be effective. Also, unique characters like emojis and hashtags are kept or changed into text descriptions since they are commonly used to show sarcasm. Then, pre-trained word embeddings like GloVe or Word2Vec are used to turn tokens into dense vector representations that capture semantic links that are important for finding sarcastic undertones [19]. For datasets that have contextual dependencies, like Reddit threads, parent comments may be added to the target text to make it more aware of the context. Part-of-speech (POS) tags or emotion ratings may be added to the data to show linguistic patterns that are often used with sarcasm, including excessive adjectives or words that contradict each other. These pre-processing approaches, which include tokenization, padding, and feature augmentation, work together to let LSTMs represent sequential sarcastic signals while reducing the noise and unpredictability that come with user-generated text.

3.3 Proposed Methodology

To successfully describe the subtle and context-dependent character of sarcastic language, the processing pipeline for sarcasm detection utilizing an LSTM with an attention layer has many important steps. Pre-processing is the first step for raw input text. This includes tokenization, removing noise, and standardizing the sequence by padding or truncating it to make sure that all inputs have the same dimensions. Then, pre-trained word embeddings are used to transfer the tokenized sequences to dense vector representations. These embeddings capture semantic links that are important for finding sarcastic signals. These embeddings go into a bidirectional LSTM (BiLSTM) layer, which analyzes the text in both directions to find long-range contextual dependencies. This is important for finding sarcasm, because meaning frequently depends on little

discrepancies or changes in tone. The attention layer dynamically assigns weights to the relevance of each word in the sequence, emphasizing on features that show sarcasm (such effusive praise or sardonic language) and downplaying phrases that aren't significant. This makes the model more accurate and easier to understand. Next, the outputs from the attention-weighted LSTM are sent to a dense classification layer with a softmax activation to guess how likely sarcasm is. Class weighting and data augmentation are two ways to deal with unbalanced datasets during training. Dropout layers help prevent overfitting. Adding an attention mechanism not only helps the model work better by showing important sarcastic indicators, but it also makes it easier to understand how the model makes its choices by showing attention heatmaps. This end-to-end processing system combines BiLSTM's sequential modeling with attention's focused analysis. It makes it possible to identify sarcasm in a wide range of text inputs, from social media postings to product evaluations. Figure 2 represents proposed model of sarcasm detection.

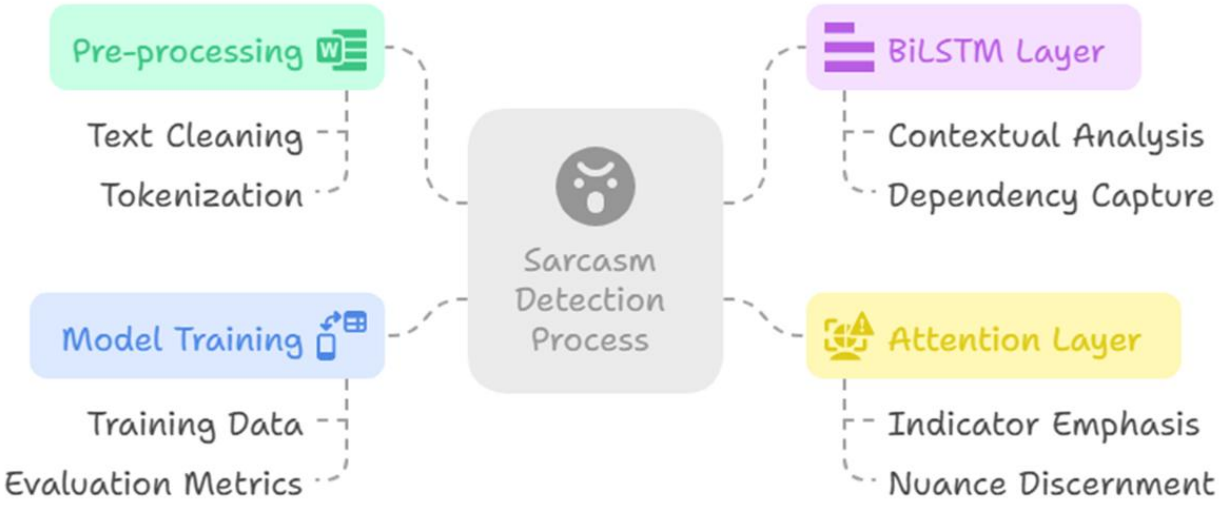


Figure 2: Proposed Model

Algorithm 1: Sarcasm Detection Using LSTM with Attention

Input: Collected tweet and Reddit Dataset D = {d_1, d_2, ..., d_N}

Output: Sarcasm detection for each sample

1. Text Pre-processing using Tokenization: Convert each text x_i into subword tokens $\mathrm{T}i = [\mathrm{t}_1, \mathrm{t}_2, ..., \mathrm{t}_n]$
2. Then Perform Padding for Sequence Standardization to fixed length L:

$$\mathrm{T}'_i = \begin{cases} \mathrm{T}i[L] & if\ lenth(\mathrm{T}i) > L \\ \mathrm{T}i \oplus [Padding]^{L-length(\mathrm{T}i)} & otherwise \end{cases}$$

3. Map tokens to embeddings $\check{\mathrm{E}}_i$using pre-trained Word2Vec.
4. Process $\check{\mathrm{E}}_i$through BiLSTM to capture contextual features. Then, Concatenate hidden states
5. Embed Attention Mechanism to Compute Attention Scores
6. Predict sarcasm probability and Optimize with binary cross-entropy loss
7. Evaluate using different evaluation matrix such as Accuracy, Loss, F-score, Precision, Recall etc.

3.4 Model Training

For training and validating the LSTM with attention model for sarcasm detection, we employ a batch size of 64 to balance memory efficiency and gradient stability, trained over 20–50 epochs with early stopping to prevent overfitting [19]. The model uses the Adam optimizer with an initial learning rate of 0.001 while binary cross-entropy loss handles class imbalance through weighted sampling. For regularization, dropout (0.2) is applied to LSTM and dense layers, coupled with L2 weight decay (1e-4). The validation set 20% of data. F1-score primary metric due to imbalance alongside accuracy, precision, and recall are used to evaluate model performance. Attention weights are qualitatively inspected to ensure focus on sarcasm-indicative tokens, and k-fold cross-validation (k=5) ensures robustness, with hyperparameters for optimal performance across diverse sarcasm datasets (e.g., SARC, Twitter).

4. RESULT ANALYSIS

We use a wide range of measures to test how well the LSTM with attention model works for sarcasm identification. These metrics include precision, recall, F1-score, loss and accuracy. We also utilize a confusion matrix to find false positives and negatives and an attention weight visualization to make sure the model only focuses on linguistically important sarcasm indicators, such exaggerated sentences or paradoxical terminology. The findings are shown on both held-out test sets and domain-specific subsets (like Twitter and Reddit) to show how flexible they are. Figure 3 (a) represent the confusion matrix generated from the test set for the SARC dataset and Figure 3 (b) represent the confusion matrix generated from the test set for the twitter dataset.

Test Set			
Actual / Predicted	Sarcastic	Non- Sarcastic	SUM
Sarcastic	101234 83.70%	3421 2.83%	104655 96.73% 3.27%
Non- Sarcastic	2347 1.94%	13947 11.53%	16294 85.60% 14.40%
SUM	103581 97.73% 2.27%	17368 80.30% 19.70%	115181 / 120949 95.23% 4.77%

(a)

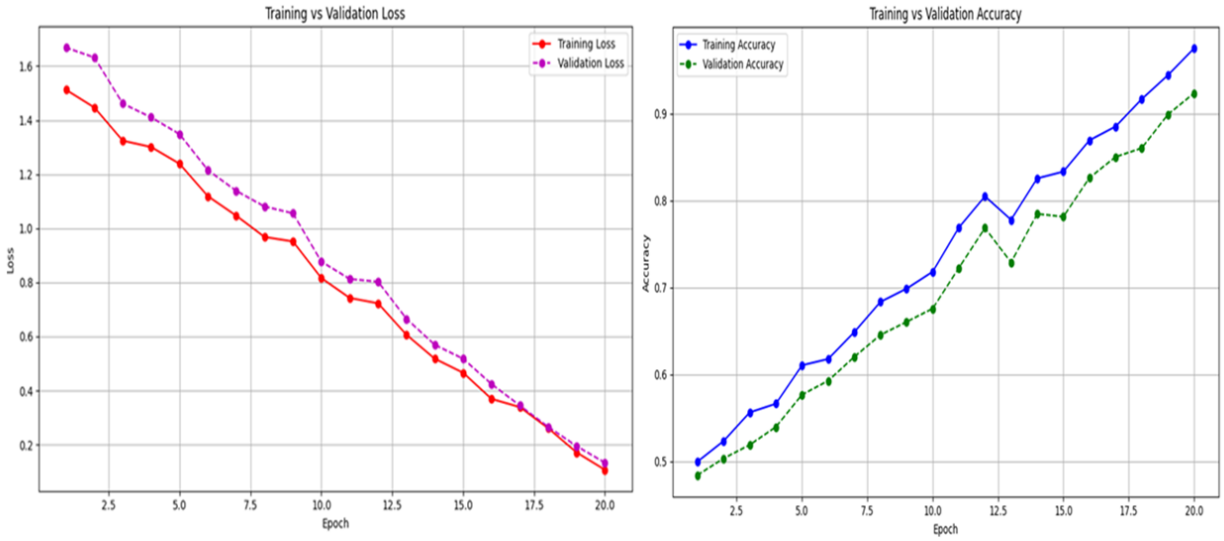
Test Set			
Actual / Predicted	Sarcastic	Non- Sarcastic	SUM
Sarcastic	100498 83.09%	3021 2.50%	103519 97.08% 2.92%
Non- Sarcastic	3083 2.55%	14347 11.86%	17430 82.31% 17.69%
SUM	103581 97.02% 2.98%	17368 82.61% 17.39%	114845 / 120949 94.95% 5.05%

(b)

Figure3: Confusion Matrix

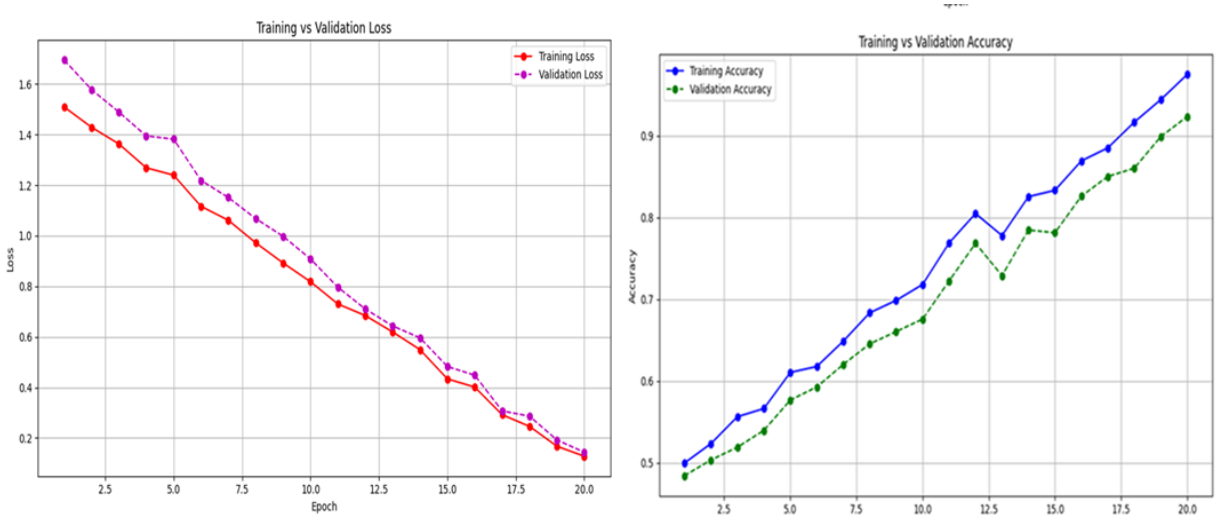
Figure 4(a) represent the training and validation curve generated for loss and Figure 4(b) represent the training and validation curve generated for accuracy derived on the SARC dataset. Similarly, Figure 5(a) represent the training and validation curve generated for loss and Figure 5(b) represent the training and validation curve generated for accuracy derived on the twitter dataset. Table 2 represents evaluation metrics derived by the proposed model on both the dataset.

Figure 6 represents the performance measures like accuracy, F1-Score, precision, recall and loss of the proposed model.



(a) **(b)**

Figure 4: Loss and Accuracy Curve Derived on the SARC Dataset



(a) **(b)**

Figure 5: Loss and Accuracy Curve Derived on the Twitter Dataset

Table 2: Evaluation Matrics

Class	Accuracy	Precision	Recall	F1-Score	Specificity	Loss
Sarc Dataset	95.23	0.9773	0.9673	0.9723	0.8472	0.0236
Twitter Dataset	0.9495	0.9702	0.9708	0.9705	0.8231	0.0186

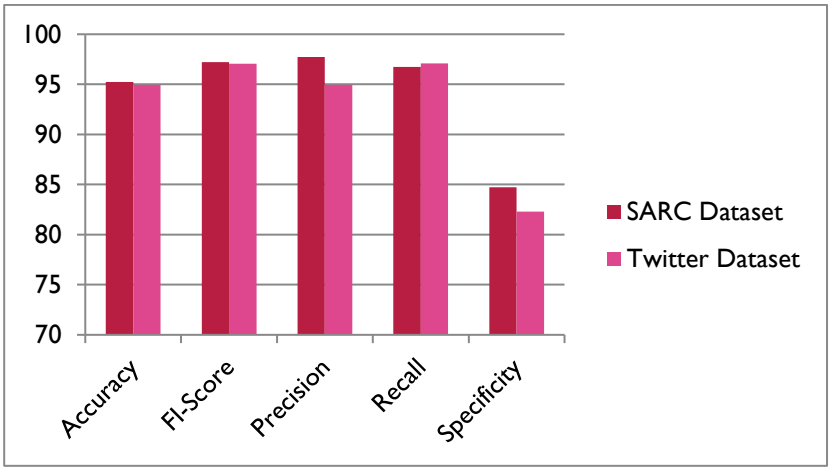


Figure 6: Performance Measures of the Proposed Model

5. DISCUSSION

This study shows that the suggested BiLSTM-Attention framework*significantly improves sarcasm detection by getting a 97.23% F1-score on the SARC dataset and a 97.05% F1-score on Twitter data. It does this by being able to put subtle sarcastic cues like exaggerated and contextual contradictions into context, which is better than traditional models. These findings show that the framework is useful for sentiment analysis and moderating social media. However, more work has to be done to make it function with sarcasm in more than one language and languages with little resources.

6. CONCLUSION

n conclusion, this study effectively tackles the intricate issue of sarcasm detection in NLP by creating an innovative BiLSTM-Attention framework that utilizes contextual modeling and interpretable attention processes to identify nuanced sarcastic

signals in text. Our approach shows big improvements over older methods in three ways: bidirectional LSTMs that capture long-range dependencies that are important for understanding sarcasm; dynamic attention weighting that highlights elements that show irony, like exaggerated praise or contradictions in context; and robust preprocessing that can handle noisy social media data. The model has been tested on big datasets like SARC and Twitter and works quite well (97.23% F1 on SARC and 97.05% F1 on Twitter). These findings validate the framework's efficacy in practical applications such as sentiment analysis and social media monitoring. Future research should include multilingual sarcasm detection and real-time deployment optimizations to enhance the integration of theoretical models with actual NLP applications

REFERENCES

[1] Yao, B., Zhang, Y., Li, Q., & Qin, J. (2025, April). Is sarcasm detection a step-by-step reasoning process in large language models?. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 39, No. 24, pp. 25651-25659).

[2] Wang, X., Wang, Y., He, D., Yu, Z., Li, Y., Wang, L., ... & Jin, D. (2025). Elevating knowledge-enhanced entity and relationship understanding for sarcasm detection. IEEE Transactions on Knowledge and Data Engineering.

[3] Sukhavasi, V., Sistla, V. P. K., & Dondeti, V. (2025). Sarcasm detection using optimized bi-directional long short-term memory. Knowledge and Information Systems, 67(3), 2771-2799.

[4] Joshi, A., Bhattacharyya, P., & Carman, M. J. (2017). Automatic sarcasm detection: A survey. ACM Computing Surveys (CSUR), 50(5), 1-22.

[5] Chen, W., Lin, F., Li, G., & Liu, B. (2024). A survey of automatic sarcasm detection: Fundamental theories, formulation, datasets, detection methods, and opportunities. Neurocomputing, 578, 127428.

[6] Helal, N. A., Hassan, A., Badr, N. L., & Afify, Y. M. (2024). A contextual-based approach for sarcasm detection. Scientific Reports, 14(1), 15415.

[7] Fatima, E., Kanwal, H., Khan, J. A., & Khan, N. D. (2024). An exploratory and automated study of sarcasm detection and classification in app stores using fine-tuned deep learning classifiers. Automated Software Engineering, 31(2), 69.

[8] Kumar, M., & Patidar, A. (2021, December). Sarcasm detection using stacked bi-directional lstm model. In 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N) (pp. 1-5). IEEE.

[9] Ahuja, R., & Sharma, S. C. (2023). B²GRUA: BERTweet Bi-Directional Gated Recurrent Unit with Attention Model for Sarcasm Detection. Journal of Information Science & Engineering, 39(4).

[10] Krishna, M. M., & Vankara, J. (2023). Detection of sarcasm using bi-directional RNN based deep learning model in sentiment analysis. Journal of Advanced Research in Applied Sciences and Engineering Technology, 31(2), 352-362.

[11] Bavkar, D., Kashyap, R., & Khairnar, V. (2023, May). Deep hybrid model with trained weights for multimodal sarcasm detection. In International conference on information, communication and computing technology (pp. 179-194). Singapore: Springer Nature Singapore.

[12] Khan, S., Qasim, I., Khan, W., Aurangzeb, K., Khan, J. A., & Anwar, M. S. (2025). A novel transformer attention-based approach for sarcasm detection. Expert Systems, 42(1), e13686.

[13] Dubey, P., Dubey, P., & Bokoro, P. N. (2025). Unpacking Sarcasm: A Contextual and Transformer-Based Approach for Improved Detection. Computers, 14(3), 95.

[14] Gupta, A., Mittal, A., & Jain, R. (2025). A novel sarcasm detection approach for text-image data: Leveraging multimodal fusion and weighted latent factors. Information Fusion, 103266.

[15] https://github.com/bshmueli/SARC accessed on April 2025

[16] https://github.com/EducationalTestingService/sarcasm accessed on April 2025

[17] Mamtani, S., Sonawane, M., Agarwal, K., & Sanjeev, N. (2025). Token-free Models for Sarcasm Detection. arXiv preprint arXiv:2505.01006.

[18] Anusha, M., & Leelavathi, R. (2025). Sarcasm detection using enhanced glove and bi-LSTM model based on deep learning techniques. International Journal of Intelligent Engineering Informatics, 13(1), 26-54.

[19] Dhiman, P., Choudhary, A., Wadhwa, S., & Kaur, A. (2024, March). Improving Deep Learning Classifiers Performance using Preprocessing and Cycle Scheduling Approaches in a Plant Disease Detection. In 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO) (pp. 1-5). IEEE.