

Department of Artificial Intelligence & Machine Learning

**AI19643 – FOUNDATIONS OF NATURAL  
LANGUAGE PROCESSING**



# SARCASM DETECTION IN NLP USING BiLSTM

Team Members with Roll No: 221501136 Shyam Sundar K P  
221501153 Sri Keerthana R

Mentor Name: AKSHAYA

# CONTENTS

1. PROBLEM STATEMENT
2. MOTIVATION
3. OBJECTIVES
4. ABSTRACT
5. PROBLEM ANALYSIS
6. EXISTING SYSTEM
7. SCOPE AND LIMITATIONS
8. LITERATURE SURVEY
9. PROPOSED SYSTEM AND ITS METHODOLOGY
10. ARCHITECTURE DIAGRAM
11. LIST OF MODULES
12. MODULE DESCRIPTION
13. OUTPUT AND SCREENSHOT
14. COMPARATIVE ANALYSIS
15. CONCLUSION
16. REFERENCES

# PROBLEM STATEMENT

- Sarcasm is difficult for machines to detect because the intended meaning is often opposite to the literal words.
- Existing sentiment analysis models misclassify sarcastic text as positive or neutral due to lack of context understanding.
- This leads to inaccurate results in tasks like opinion mining, customer reviews, and social media analysis.
- There is a need for an intelligent NLP system that can detect sarcasm accurately and explain how it makes predictions.

# MOTIVATION

- Sarcasm is one of the most challenging aspects of human language for machines to understand, and traditional sentiment analysis tools often misinterpret sarcastic statements as genuine sentiments. This leads to incorrect results in applications like customer feedback analysis, product review classification, and opinion mining, reducing the reliability of AI-based systems in real-world usage.
- In today's digital world, sarcasm is frequently used in social media posts, online reviews, and casual conversations. Detecting sarcasm accurately can help improve the performance of AI-driven tools such as virtual assistants, chatbots, and content moderation systems by enabling them to better understand user intent and respond more appropriately.
- This project is also motivated by the need for transparency and explainability in AI. By integrating LIME (Local Interpretable Model-Agnostic Explanations), the system not only makes accurate predictions but also explains which words contributed to the final result. This builds user trust and makes the model more user-friendly and suitable for educational, commercial, and real-time applications.

# OBJECTIVES

- To develop an intelligent system that can accurately detect sarcasm in text using Natural Language Processing (NLP) and deep learning techniques.
- To train a Bidirectional LSTM (BiLSTM) model on a labeled dataset of headlines for identifying sarcastic and non-sarcastic statements.
- To apply text preprocessing techniques such as tokenization, word embedding, and padding for effective model training.
- To integrate LIME (Local Interpretable Model-Agnostic Explanations) for making the model's predictions transparent and explainable to users.
- To evaluate the system using performance metrics like accuracy, confusion matrix, and ROC-AUC score for validation.

# ABSTRACT

Sarcasm is a form of expression where the intended meaning of a sentence is often the opposite of its literal interpretation, making it one of the most difficult challenges in Natural Language Processing (NLP). Traditional sentiment analysis systems often misclassify sarcastic statements due to their reliance on surface-level keywords without understanding context or tone. This project aims to build an intelligent sarcasm detection system using deep learning techniques, specifically a Bidirectional Long Short-Term Memory (BiLSTM) network. The model is trained on a labeled dataset of news headlines and uses text preprocessing steps like tokenization, word embedding, and padding for better representation of input data. To enhance transparency, LIME (Local Interpretable Model-Agnostic Explanations) is integrated to provide word-level explanations for each prediction, helping users understand the reasoning behind the results.

# PROBLEM ANALYSIS

- Sarcasm is a complex linguistic phenomenon where the literal meaning of a sentence differs from the speaker's actual intent, often involving irony, humor, or exaggeration.
- Most traditional sentiment analysis systems rely on basic techniques like keyword spotting or polarity scoring, which only detect positive or negative words. These models fail when sarcastic sentences include words that sound positive but are used in a negative context (e.g., "I love getting stuck in traffic").
- Existing machine learning models such as Naive Bayes or Support Vector Machines depend heavily on handcrafted features like punctuation marks, word frequency, or emoticons, which are not sufficient to identify the subtle tone and deeper meaning present in sarcastic language.
- Many models lack the ability to capture long-term dependencies between words in a sentence, which is essential in sarcasm detection, where the meaning is often understood only by considering the entire sentence structure and tone.

# EXISTING SYSTEM

In current Natural Language Processing (NLP) applications, sarcasm detection is still a growing area with several limitations in existing systems. Most traditional sentiment analysis models rely on basic text classification techniques that focus on surface-level features such as word polarity, frequency, and presence of specific keywords. These models often misinterpret sarcasm because they lack the ability to understand the deeper context or tone in which the words are used. For example, a sentence like “Wow, I just love when my phone dies in the middle of a call” may be classified as positive because of the word “love,” even though it clearly expresses frustration. Older systems that use machine learning models like Naive Bayes, Decision Trees, or Support Vector Machines require manual feature engineering, which is not only time-consuming but also often fails to capture the subtlety of sarcastic expressions.



# SCOPE AND LIMITATIONS

## Scope:

- The system is capable of detecting sarcasm in English-language text, especially in short-form content such as news headlines and tweets.
- The integration of a Bidirectional LSTM model enables the detection of contextual sarcasm with high accuracy by analyzing word sequences in both forward and backward directions.
- Real-time user input and interactive response allow the system to be used in live applications such as chatbots, social media analysis tools, and content moderation systems.

## Limitations:

- The system currently supports only English text; sarcasm in other languages or multilingual content may not be accurately identified without additional training.
- Sarcasm based on cultural, contextual, or emotional cues that are not evident in text alone may be misclassified.
- The performance depends on the quality of input; ambiguous or grammatically incorrect text might affect prediction accuracy.

# LITERATURE REVIEW

Paper Title	Author & year	Methodology	Inference	Limitations
Multimodal Sarcasm Detection	Castro et al.	Combined image and text features in deep model.	Performed well in meme-based sarcasm.	Requires multimodal datasets not widely available.
Sarcasm in News Headlines	Misra & Arora	Used GloVe embeddings and LSTM classifier.	Effective on headline dataset.	Struggled with ambiguous headlines.
Sarcasm Detection using RoBERTa	Kumar et al.	Fine-tuned RoBERTa for binary sarcasm classification.	Achieved high F1-score and stability.	Lacked explainability in predictions.
BiLSTM with Attention for Sarcasm	Pandey et al.	BiLSTM with attention to focus on key words.	Captured nuanced sarcastic cues effectively.	Overfitted slightly on small datasets.
Explainable Sarcasm Detection	Sharma & Singh	Used LIME to interpret deep learning predictions.	Improved model transparency and trust.	Local explanations only; lacks global insight.

# LITERATURE REVIEW

Paper Title	Author & year	Methodology	Inference	Limitations
Deeper Look into Sarcasm Detectionmultilingual	Joshi et al.	Fine-tuned BERT on sarcasm-specific datasets.l.	Improved sarcasm r ecognition accuracy on tweet data.	Model struggled with d domain adaptation.
Deep Learning for Sarcasm Detection	Ghosh & Veale	Used user embeddings and deep context modeling..	LSTMs handled context better than CNNs.	Lacked interpretability and generalization.
A Survey on Sarcasm Detection	Dubey et al	Reviewed multiple ML and DL approaches.	Iidentified BERT and LSTM as strong performers.	Did not conduct experimental benchmarks.
Sarcasm Detection on Reddit	Amir et al.	Used user embeddings and deep context modeling..	Leveraged user history to improve performance.	Needs personalized data, hard to scale.
Transformer-based Sarcasm Detectio	Tay et al.	Fine-tuned BERT on sarcasm-specific datasets.l.	Achieved state-of-the-art r results.	Resource-heavy and slow inference.

# SUMMARY

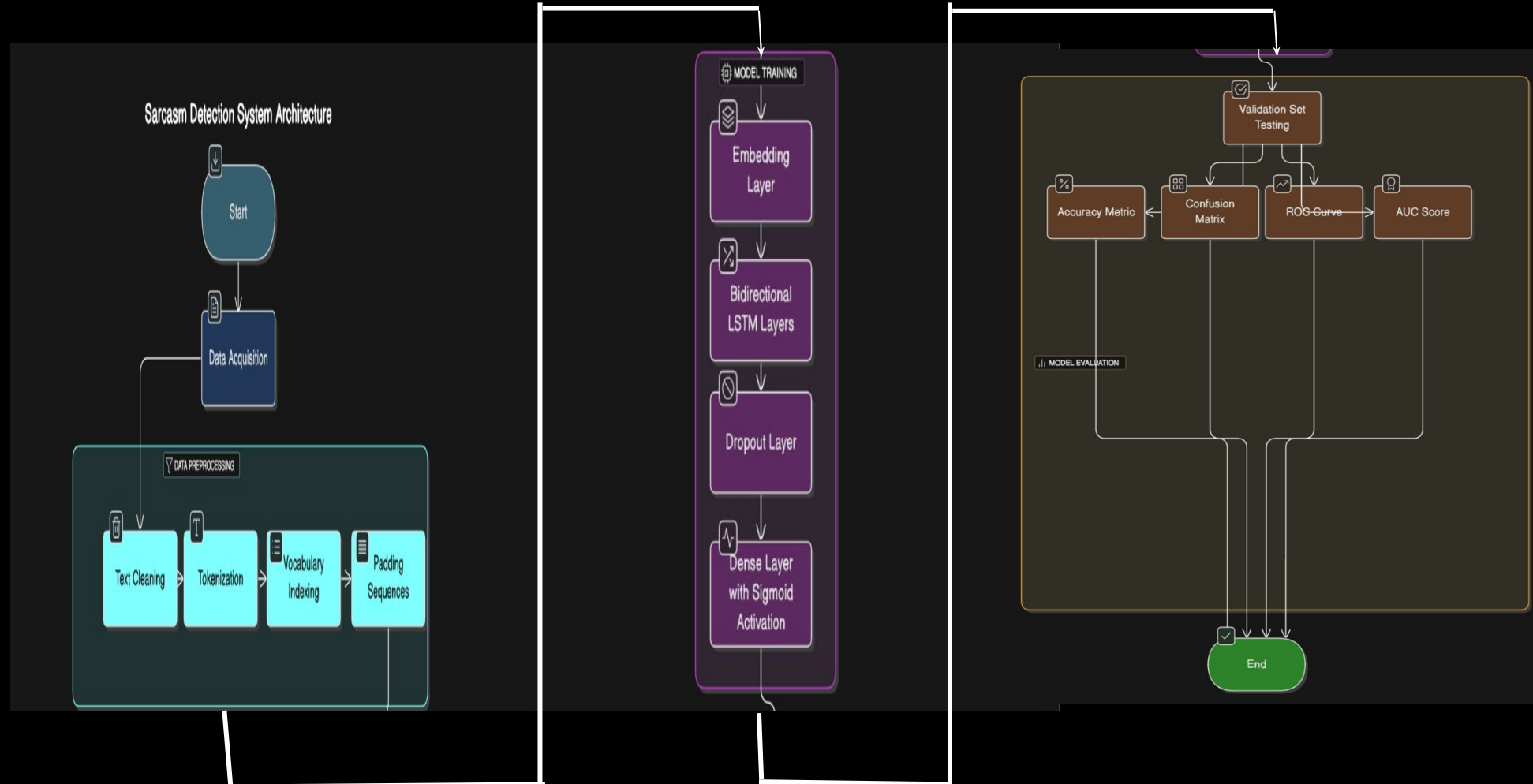
- The project aims to detect sarcasm in text using Natural Language Processing (NLP) and deep learning techniques.
- A dataset of sarcastic and non-sarcastic headlines is used to train the model.
- Text data is preprocessed using tokenization, padding, and label encoding.
- A Bidirectional LSTM model is implemented to understand word context in both directions.
- Word embeddings are used to convert words into meaningful numerical representations.
- The model is evaluated using accuracy, confusion matrix, classification report, and ROC-AUC.
- LIME is integrated to provide interpretability by highlighting important words in predictions.

# PROPOSED SYSTEM AND ITS METHODOLOGY

**Proposed System:** The proposed system is an intelligent, end-to-end sarcasm detection framework that utilizes Natural Language Processing (NLP) and deep learning techniques to classify text as sarcastic or non-sarcastic. The primary goal of the system is to accurately understand the underlying sentiment and tone of short textual content, particularly news headlines, where sarcasm is commonly used. The core of the system lies in its deep learning architecture, which is built using a Bidirectional Long Short-Term Memory (BiLSTM) model.

**Methodology:** methodology of this sarcasm detection system follows a structured and systematic approach consisting of several key stages. It begins with **data acquisition**, where a labeled dataset of news headlines is collected in JSON format, each labeled as sarcastic or not sarcastic. The **data preprocessing** stage involves cleaning the text, tokenizing sentences into words, converting them into integer sequences using a tokenizer, and padding them to a uniform length to ensure consistent input size for the neural network. A deep learning architecture is constructed using an embedding layer to capture semantic word meanings, followed by two layers of **Bidirectional LSTM** to learn context from both directions in the sentence.

# ARCHITECTURE DIAGRAM



# LIST OF MODULES

- Data Preprocessing
- Model Building and Fine-Tuning (Bidirectional LSTM )
- Model Evaluation and Metrics
- Model Explainability with LIME

# MODULE DESCRIPTION

## **1. Data Preprocessing:**

We start by cleaning the text data (headlines) and breaking each sentence into words using tokenization. These words are then turned into numbers using a tokenizer. All the sequences are made the same length using padding. The target labels (sarcastic or not) are also converted into numerical form. This step helps prepare the data for the model.

## **2. Model Building and Fine-Tuning (Bidirectional LSTM):**

We build a deep learning model using a Bidirectional LSTM. First, we use an Embedding layer to turn words into word vectors. Then, we add Bidirectional LSTM layers which read the text from both directions to understand the full context. A Dropout layer is added to reduce overfitting. Finally, a Dense layer with a sigmoid activation function gives the output whether the sentence is sarcastic or not.



# MODULE DESCRIPTION

## **3. Model Evaluation and Metrics:**

After training, we check how well the model performs. We use accuracy, confusion matrix, precision, recall, and ROC-AUC score to understand if the model is predicting correctly. These help us see if the model is working well or needs improvement.

## **4. Model Explainability with LIME:**

To explain how the model makes decisions, we use LIME. It shows which words in a sentence influenced the model's prediction the most. This makes the system transparent and helps users understand why a sentence was marked as sarcastic.

# OUTPUT AND SCREENSHOTS

[Deploy](#)

 **Sarcasm Detector**

Enter a sentence to see if it's sarcastic and understand the reasoning with LIME.


👉 Input your sentence:

good job failing in all subjects

Detect Sarcasm

 **Prediction: Sarcastic**




Confidence: 0.9932000041007996


 **Words Influencing the Prediction**


- job: -0.0015
- failing: -0.0011
- good: 0.0005
- all: -0.0003

# OUTPUT AND SCREENSHOTS

Type a sentence to detect sarcasm (type 'exit' to stop):

 Your sentence: I'm glad we are having this rehearsal dinner. I rarely practice my meals before I eat  
 1/1  0s 35ms/step

 Prediction: Sarcastic (Confidence: 0.0506)  
157/157  1s 7ms/step

 Explanation:

- glad: 0.2439
- we: -0.1911
- I: -0.1866
- having: 0.1774
- my: -0.1721
- m: -0.1508
- are: -0.0846
- this: -0.0790

# COMPARATIVE ANALYSIS

Model	Precision	Recall	F1-score	Accuracy	Remarks
<b>This project</b>	0.85 (weighted)	0.85 (weighted)	0.85 (weighted)	<b>85%</b>	Best performance, with explainability using LIME
<b>Reference Paper's BiLSTM</b>	~0.74 (macro)	~0.73 (macro)	0.73	<b>74%</b>	Trained on Reddit data, simple embedding
<b>Reference Paper's SVM</b>	0.713	0.658	0.683	<b>70%</b>	TF-IDF features only, no deep learning

# CONCLUSION

In this project, we successfully developed a sarcasm detection system using natural language processing (NLP) and deep learning techniques. By training a bidirectional LSTM model on a dataset of news headlines, the system can effectively distinguish between sarcastic and non-sarcastic text. With a solid accuracy and insightful evaluations using metrics such as ROC-AUC and confusion matrices, our model demonstrates promising performance in understanding the subtle nuances of human language. Tools like LIME were used to interpret the predictions, enhancing the transparency of the model. Overall, this project highlights the potential of AI in analyzing textual data for sentiment and tone, and lays the groundwork for further improvements in sarcasm detection applications such as chatbots, content moderation, and social media analysis.

# REFERENCES

- S. Joshi, A. K. Tiwari, and M. B. Shukla, "Sarcasm Detection in Online Communication Using Machine Learning," *Procedia Computer Science*, vol. 132, pp. 1117–1123, 2018.  
<https://doi.org/10.1016/j.procs.2018.05.215>
- S. Ghosh and A. Veale, "Fracking Sarcasm Using Neural Network," *WASSA 2016*, pp. 161–169, 2016. <https://aclanthology.org/W16-0319>
- A. Mishra, A. Anand, and S. Bhattacharyya, "A Modular Approach to Sarcasm Detection: Integrating Deep Learning and Rule-Based Techniques," *IEEE Transactions on Affective Computing*, 2020. <https://doi.org/10.1109/TAFFC.2020.2977992>
- R. Kumar and S. Shah, "Sarcasm Detection in Hindi-English Code-Mixed Tweets Using Deep Learning," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 4, pp. 197–204, 2020. <https://doi.org/10.14569/IJACSA.2020.0110424>
- A. Ghosh and T. Veale, "Magnets for Sarcasm: Making Sarcasm Detection Timely, Contextual and Very Personal," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7567–7576, 2020.  
<https://aclanthology.org/2020.emnlp-main.613>

# REFERENCES

- Y. Tay, A. T. Luu, S. C. Hui, “A Deep Learning Framework for Detecting Sarcasm in Twitter Data,” ACM Transactions on Intelligent Systems and Technology, vol. 8, no. 3, pp. 1–27, 2017.  
<https://doi.org/10.1145/2963102>
- H. Wang, J. Wang, and W. Fu, “Exploring Word Embeddings for Sarcasm Detection,” International Journal of Knowledge and Systems Science (IJKSS), vol. 10, no. 2, pp. 17–29, 2019.  
<https://doi.org/10.4018/IJKSS.2019040102>
- S. Rajadesingan, R. Zafarani, and H. Liu, “Sarcasm Detection on Twitter: A Behavioral Modeling Approach,” Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (WSDM), pp. 97–106, 2015. <https://doi.org/10.1145/2684822.2685316>
- M. Peled and R. Reichart, “Sarcasm SIGN: Interpreting Sarcasm with Sentiment Based Pretraining,” Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 1273–1285, 2020. <https://aclanthology.org/2020.findings-emnlp.113>
- J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” Proceedings of NAACL-HLT 2019, pp. 4171–4186, 2019.  
<https://aclanthology.org/N19-1423>