JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A COMPARATIVE STUDY OF BILSTM AND ROBERTA MODEL FOR SARCASM DETECTION

[1]Lakshmi H R,[2]Mohammad Farhaan ,[3]Dr Mallikarjun H M

[1] Student,[2] Student,[3] Assistant Professor,

[1]Dept. of CSE (AIML), RNSIT, Bangalore,India

Abstract: Detecting sarcasm in text remains a complex challenge in Natural Language Processing (NLP) due to its subtlety and reliance on context. This research compares the effectiveness of two deep learning models—Bidirectional Long Short-Term Memory (BiLSTM) and the Robustly Optimized BERT Pretraining Approach (RoBERTa)—for identifying sarcasm in news headlines. Leveraging the Sarcasm Headlines Dataset v2, which contains 28,619 labeled samples, both models were assessed using metrics like accuracy, F1-score, and loss over several training epochs. Findings show that RoBERTa significantly outperforms BiLSTM, achieving a validation accuracy of 93.76% and an F1-score of 0.9326 at epoch 10, compared to BiLSTM's 84.70% accuracy and 0.8403 F1-score. RoBERTa's advantage stems from its transformer-based architecture and self-attention mechanisms, enabling better understanding of contextual nuances, whereas BiLSTM's sequential nature limits its grasp of long-range dependencies. The study underscores the critical role of model architecture in sarcasm detection and discusses the computational considerations involved. Future directions include exploring hybrid models, incorporating multimodal data, and applying domain-specific fine-tuning to improve detection performance further.

Keywords

Sarcasm detection, Natural Language Processing (NLP), RoBERTa, BiLSTM, Transformer models, Sentiment analysis, Deep learning, Contextual embeddings, Text classification, Comparative analysis

1. INTRODUCTION

A. Sarcasm and Its Social Impacts

Sarcasm is a type of verbal irony where the intended meaning contrasts with the literal words, typically used to convey humor or emphasis. Though it's common in everyday conversation, sarcasm can easily be misunderstood—particularly in digital or text-based communication where cues like tone, facial expressions, and situational context are missing. This lack of clarity can cause confusion, offense, or even spark conflicts online. On social media, sarcasm frequently appears in political commentary, trolling, and dark humor, posing challenges for both people and machines in accurately identifying whether a statement is sincere or sarcastic.

Sarcasm not only influences the tone of communication but also impacts public opinion, user sentiment, and discourse quality. Misunderstanding sarcasm in natural language processing (NLP) tasks—such as sentiment analysis or opinion mining—can severely skew results, especially in fields like mental health monitoring, market research, and misinformation detection.

B. Importance of Sarcasm Detection

Accurate sarcasm detection is vital for enhancing the performance of NLP systems. Conventional sentiment analysis tools often rely on individual word sentiments, leading to misinterpretations—for instance, the phrase "Oh great, another Monday!" may seem positive due to the word "great," even though its actual tone is negative. Incorporating sarcasm detection allows systems like chatbots, recommendation engines, and mental health analysis tools to better understand context, thereby improving user engagement and decision-making. Additionally, detecting sarcasm is critical for moderating online spaces, where harmful or deceptive content is frequently masked in sarcastic language to bypass content filters.

C. Motivation and Statistics

The increasing reliance on AI-driven systems in understanding user emotions and intent has highlighted the limitations of current models in detecting subtle language cues like sarcasm. Recent studies have shown that sarcasm accounts for **approximately 23%** of the misunderstandings in sentiment analysis tasks on social media platforms such as Twitter and Reddit. According to a 2022 Pew Research article, over **67%** of online users reported encountering sarcasm in comments, memes, or captions on a daily basis.

In addition, recent progress in pre-trained transformer models like RoBERTa (Robustly Optimized BERT Pretraining Approach) has demonstrated strong capabilities in understanding complex contextual cues. Unlike conventional machine learning methods that depend on manually engineered features, RoBERTa leverages large-scale text corpora to automatically learn semantic relationships. This enables it to effectively recognize sarcastic remarks, even when the literal sentiment of the text is misleading or unclear.

This research aims to explore the effectiveness of RoBERTa in detecting sarcasm and improving interpretability in sentiment-based applications. By fine-tuning RoBERTa on annotated sarcastic datasets, we aim to reduce misclassification errors and enhance the robustness of modern NLP systems.

D. Purpose of the Study

The primary objective of this study is to evaluate and compare the performance, accuracy, and efficiency of two deep learning models—BiLSTM and RoBERTa—for detecting sarcasm in news headlines. By applying both models to a labeled sarcasm dataset, this research offers a comprehensive analysis of their respective capabilities in identifying sarcastic expressions. The study focuses on how well each model performs in recognizing sarcasm, compares their accuracy using metrics such as precision, recall, and F1-score, and examines the computational demands of each model, particularly in terms of scalability to larger datasets. Through this comparative approach, the study aims to determine the more suitable model for sarcasm detection and contribute meaningful insights to the broader field of Natural Language Processing (NLP).

E. Motivation

This research is driven by the growing need for NLP systems that can accurately interpret subtle and context-dependent language constructs such as sarcasm. Understanding sarcasm is essential for enhancing various applications, including sentiment analysis, content moderation, and customer support. In sentiment analysis, accurate sarcasm detection prevents the misinterpretation of ironic statements, which could otherwise skew results. In the context of content moderation, it helps avoid the misclassification of sarcastic comments as harmful or offensive. Similarly, in customer service interactions, recognizing sarcasm enables automated systems to respond more appropriately, especially when users express frustration or humor. By comparing BiLSTM and RoBERTa, this study aims to explore how different model architectures handle the complexity of sarcasm in short, often ambiguous texts like news headlines.

2. LITERATURE REVIEW

A. Survey

This study emphasizes the importance of multimodal cues, such as audiovisual features, in sarcasm detection. The authors introduce the MUStARD dataset, derived from TV shows, and show that multimodal approaches improve sarcasm classification accuracy by reducing error rates by 12.9%.[1]

Sarcasm detection is crucial for sentiment analysis but remains challenging due to its contradictory nature. This study compares BERT and LSTM for sarcasm identification, demonstrating BERT's superiority in improving sentiment accuracy, particularly on Twitter data.[2]

This research presents a sarcasm detection system using various models and data augmentation techniques. The RoBERTa-based model with mutation-based augmentation achieved an F1-score of 0.414, showcasing the benefits of transformer models in sarcasm classification.[3]

A BERT-based model is proposed to capture intra- and inter-modality incongruity for multimodal sarcasm detection. Using self-attention and co-attention mechanisms, the model outperforms existing approaches on a multimodal sarcasm detection dataset.[4]

Results indicate higher accuracy in Indonesian (88.33% for balanced data) compared to English (79%), highlighting the model's effectiveness in cross-linguistic contexts.[5]

Evidential deep learning, combined with LSTM and GRU, is applied for sarcasm detection in news headlines. The approach effectively models sequential data, capturing sentiments and contextual dependencies within news articles.[6]

An LSTM with an attention mechanism (LSTM-AM) achieves 99.86% accuracy in sarcasm and irony detection on social media. The study explores transfer learning and multimodal integration (text, emojis, images) to improve sentiment analysis.[7]

Sarcasm detection is treated as a binary classification problem, with a multi-head attention-based BiLSTM (MHA-BiLSTM) model outperforming feature-rich SVM models. The attention mechanism enhances BiLSTM's performance in detecting sarcasm.[8]

This research focuses on sarcasm detection in news headlines using a BERT-LSTM model. The BERT-processed word vectors improve the LSTM classification, achieving high recognition accuracy on professional news datasets.[9]

A BERT-based approach integrated with fuzzy logic is proposed to enhance sarcasm detection. Fuzzy logic improves precision by accommodating ambiguity and context dependencies, refining NLP models for better sarcasm recognition.[10]

Multimodal sarcasm detection is explored by integrating text and images to enhance classification performance. Given the rise of social media content, this study tells the importance of considering multimodal data for sarcasm identification.[11]

A hybrid CNN-LSTM model has been proposed for sarcasm detection, combining convolutional layers to extract local textual features with LSTM layers to capture sequential dependencies. This integrated architecture demonstrates superior performance compared to conventional machine learning techniques in the context of sentiment analysis.12]

Sarcasm detection is improved using hybrid deep learning architectures, combining CNNs, LSTMs, and attention mechanisms. The approach enhances contextual understanding and achieves better results than standalone deep learning models.[13]

The study investigates multi-feature fusion techniques for sarcasm detection, integrating lexical, syntactic, and contextual features. Results indicate that feature fusion improves sarcasm classification compared to single-feature models.[14]

Transformer-based architectures, including BERT and GPT, are evaluated for sarcasm detection. The study finds that pre-trained models significantly enhance sarcasm identification by capturing contextual nuances.[15]

A novel sarcasm detection model leverages adversarial learning and contrastive loss functions. The approach refines sarcasm classification by minimizing feature overlap between sarcastic and non-sarcastic expressions.[16]

Sentiment-aware sarcasm detection is explored using graph neural networks (GNNs). The study demonstrates that GNNs effectively capture relational dependencies, improving sarcasm classification accuracy.[17]

This research utilizes reinforcement learning for sarcasm detection, optimizing classifier decision-making. The approach adapts to dynamic language patterns, improving detection performance over static models.[18]

Multitask learning is applied to sarcasm detection, leveraging shared knowledge across related NLP tasks. The approach enhances model generalization, improving sarcasm classification across diverse datasets.[19]

This paper introduces a fusion model combining symbolic AI with deep learning for sarcasm detection. The hybrid approach effectively integrates rule-based reasoning with neural network-based feature extraction.[20]

Ensemble learning methods, including voting classifiers and stacking, are explored for sarcasm detection. Results indicate that ensemble models outperform individual classifiers in sarcasm recognition tasks.[21]

A contrastive learning framework is proposed for sarcasm detection, enhancing feature representation through supervised contrastive loss. The approach significantly improves sarcasm classification by capturing subtle linguistic variations.[22]

B. Understanding from Review

Paper ID	Task Performed	Technology Used	Advantages	Limitations
1	Created MUStARD dataset for multimodal sarcasm detection	Audiovisual utterances, Contextual data from TV shows	Improved sarcasm detection by 12.9% in F-score	Limited to scripted dialogues, lacks real-world diversity
2	Compared BERT and LSTM for sarcasm detection	BERT, LSTM	BERT improves sentiment accuracy in Twitter interactions	Dependent on dataset quality, struggles with highly nuanced sarcasm
3	Evaluated models for sarcasm detection in SemEval-2022	RoBERTa, Data Augmentation	Achieved F1-sarcastic of 0.414 after improvements	Initial results were lower, dependency on dataset tuning
4	Proposed BERT-based model with inter-modality incongruity detection	BERT, Self-attention, Co-attention	State-of-the-art performance on multi-modal sarcasm dataset	Complexity increases with multimodal data processing
5	Used paragraph2vec for sarcasm context extraction	Paragraph2vec, LSTM	High accuracy (88.33% in Indonesian, 79% in English)	Lower performance with imbalanced data (76.66% in Indonesian, 54.5% in English)
6	Applied Evidential deep learning to news sarcasm detection	Evidential DL, LSTM, GRU	Effective in analyzing sarcasm in news headlines	Complexity in estimating uncertainty
7	Used LSTM with Attention Mechanism (LSTM-AM) for sarcasm detection	LSTM-AM, Transfer Learning	Achieved 99.86% accuracy on social media text	Needs more real-world testing to confirm generalizability
8	Developed Multi-Head Attention BiLSTM (MHA-BiLSTM) model	MHA-BiLSTM, Feature-rich SVM	Multi-head attention enhances BiLSTM performance	Requires significant feature engineering
9	Proposed Bert-LSTM for sarcasm detection in news headlines	BERT, LSTM	High recognition rate in professional news dataset	Needs more evaluation on social media sarcasm
10	Integrated fuzzy logic with BERT for sarcasm detection	Fuzzy logic, Transformer, BERT, RNN	Fuzzy logic enhances sarcasm detection by accommodating ambiguity	Computationally intensive
11	Developed multimodal sarcasm detection model	RoBERTa, Image-text fusion	Leverages multimodal data for better accuracy	Handling images and text together is computationally expensive
12	Used CNN-BiLSTM for sarcasm detection	CNN, BiLSTM	Captures both spatial and sequential features	CNN can struggle with highly ambiguous sarcasm
13	Proposed sarcasm detection in dialogue context	Contextual embeddings, LSTM	Considers past dialogue for better sarcasm understanding	Dependent on quality of conversational dataset
14	Analyzed emoji influence on sarcasm detection	Emoji embeddings, CNN	Emojis improve sarcasm detection accuracy	Limited to text containing emojis
15	Used attention-based LSTM for sarcasm detection	Attention mechanism, LSTM	Attention mechanism enhances contextual sarcasm detection	May not generalize to unseen sarcastic expressions
16	Compared transformer-based models for sarcasm detection	BERT, RoBERTa, XLNet	RoBERTa outperforms other transformers	Transformers are resource-intensive
17	Investigated sarcasm detection with GNN	Graph Neural Networks (GNNs)	Captures sarcasm in user interactions and relationships	Requires social network structure for best performance
18	Proposed reinforcement learning approach for sarcasm detection	Reinforcement Learning (RL), Policy Gradient	Adapts dynamically to new sarcasm patterns	Needs continuous training for effectiveness
19	Introduced sarcasm-aware sentiment classification	Sentiment analysis, BERT, Sarcasm-aware embeddings	Improves sentiment classification by removing sarcasm bias	Limited by sarcasm detection accuracy
20	Developed sarcasm detection in code-mixed languages	Code-mixed dataset, LSTM, Embeddings	Handles multiple languages effectively	Requires large multilingual dataset
21	Used GANs for sarcasm generation and detection	Generative Adversarial Networks (GANs), BERT	GANs improve sarcastic text generation and training	Training instability in GANs
22	Proposed hybrid sarcasm detection model	Hybrid Model (DL + Lexical Features)	Combines rule-based and deep learning methods for better performance	Feature engineering required for best results

3. METHODOLOGY

A. Dataset

Dataset Name: Sarcasm Headlines Dataset v2

Source: This dataset was originally compiled from two primary news outlets:

The Onion (a satirical news site, used as the source for sarcastic headlines)

The Huffington Post (a real news site, used as the source for non-sarcastic headlines)

Size: The dataset contains **28619 entries**.

Structure:

Each entry in the dataset is a JSON object with the following fields:

"headline": A string containing the news headline.

"is_sarcastic": A binary label where 1 indicates sarcasm and 0 indicates non-sarcasm.

"article_link": A URL pointing to the full article.

Labeling Method:

Headlines from *The Onion* were labeled as **sarcastic (is_sarcastic = 1)**.

Headlines from *The Huffington Post* were labeled as **non-sarcastic (is_sarcastic = 0)**.

This labeling assumes source-based sarcasm detection, relying on the inherent nature of the publishing platform.

Data Collection Process:

The main objective of the dataset is to serve as a standardized benchmark for evaluating sarcasm detection models developed using Natural Language Processing and machine learning methods.

{"is_sarcastic": 1, "headline": "thirtysomething scientists unveil doomsday clock of hair loss", "article_link": "https://www.theonion.com/thirtysomething-scientists-unveil-doomsday-clock-of-hai-1819586205"}
{"is_sarcastic": 0, "headline": "dem rep. totally nails why congress is falling short on gender, racial equality", "article_link": "https://www.huffingtonpost.com/entry/donna-edwards-inequality_us_57455f7fe4b055bb1170b207"}
{"is_sarcastic": 0, "headline": "eat your veggies: 9 deliciously different recipes", "article_link": "https://www.huffingtonpost.com/entry/eat-your-veggies-9-delici_b_8899742.html"}
{"is_sarcastic": 1, "headline": "inclement weather prevents liar from getting to work", "article_link": "https://local.theonion.com/inclement-weather-prevents-liar-from-getting-to-work-1819576031"}
{"is_sarcastic": 1, "headline": "mother comes pretty close to using word 'streaming' correctly", "article_link": "https://www.theonion.com/mother-comes-pretty-close-to-using-word-streaming-cor-1819575546"}
{"is_sarcastic": 0, "headline": "my white inheritance", "article_link": "https://www.huffingtonpost.com/entry/my-white-inheritance_us_59230747e4b07617ae4cbe1a"}
{"is_sarcastic": 0, "headline": "5 ways to file your taxes with less stress", "article_link": "https://www.huffingtonpost.com/entry/5-ways-to-file-your-taxes_b_6957316.html"}
{"is_sarcastic": 1, "headline": "richard branson's global-warming donation nearly as much as cost of failed balloon trips", "article_link": "https://www.theonion.com/richard-bransons-global-warming-donation-nearly-as-much-1819568749"}
{"is_sarcastic": 1, "headline": "shadow government getting too large to meet in marriott conference room b", "article_link": "https://politics.theonion.com/shadow-government-getting-too-large-to-meet-in-marriott-1819570731"}
{"is_sarcastic": 0, "headline": "lots of parents know this scenario", "article_link":

Figure 1: "Sarcasm_Headlines_Dataset_v2.json" dataset

B. Data Preprocessing

The dataset underwent different preprocessing techniques for the Bi-LSTM and RoBERTa models due to the distinct nature of their architectures.

Bi-LSTM Preprocessing

For the Bi-LSTM model, extensive manual text preprocessing was applied to prepare the raw headline text. First, all headlines were converted to lowercase to ensure uniformity and reduce vocabulary size. Line breaks (\n) were removed to clean the text further. The headlines were then tokenized into individual words using Python's split() function. Each word was lemmatized using the WordNetLemmatizer from the NLTK library, converting words to their base or dictionary form to reduce inflectional variation. Stopwords, such as "the", "is", and "and", along with punctuation marks, were removed to eliminate irrelevant tokens that do not contribute significantly to semantic meaning. The remaining meaningful words were rejoined into cleaned sentences.

These preprocessed sentences were then tokenized into numerical sequences using the Tokenizer class from Keras, with a vocabulary size limit and an out-of-vocabulary (OOV) token defined. After tokenization, the sequences were padded using Keras' pad_sequences function to ensure all input sequences were of uniform length. Padding was done post-sequence, and longer sequences were truncated as needed. This step is essential for training the Bi-LSTM model, which requires fixed-length input vectors.

RoBERTa Preprocessing

In contrast, the RoBERTa model employed the preprocessing functionality provided by the simpletransformers library, which automates all required transformations in line with the model's architecture. The raw text was passed directly to the RoBERTa model, where it was tokenized using Byte-Pair Encoding (BPE) through RoBERTa's native tokenizer. This tokenizer handled subword tokenization, added special tokens such as <s> (start of sentence) and </s> (end of sentence), and converted the tokens into input IDs as expected by the model. Unlike Bi-LSTM preprocessing, no manual lemmatization, stopword removal, or lowercasing was applied, as RoBERTa is case-sensitive and was pre-trained with casing intact. Additionally, attention masks were automatically generated, and sequences were padded or truncated to the specified maximum sequence length during training.

This dual approach to preprocessing ensured that each model type received inputs tailored to its design—manual preprocessing for Bi-LSTM and model-specific automated preprocessing for RoBERTa.

C. Model Architecture:

The Bi-LSTM model is structured to leverage the benefits of both convolutional and sequential processing layers for sarcasm detection. It includes several key components as follows.

Raw text is converted into tokens (words, subwords, or characters), followed by padding to standardize the length of all sequences. Padding ensures uniform input dimensions by adding special tokens (like zeros) to shorter sequences.

The embedding layer transforms the tokenized words into dense vectors of fixed size, capturing the semantic meaning of words. The model learns these embeddings during training, allowing it to better understand the relationships between words in context.

Dropout is employed as a regularization technique to combat overfitting. It randomly sets a fraction of input units to zero during training, preventing the model from relying too heavily on specific features. Dropout rates, typically between 0.2 and 0.5, are tuned for optimal performance.

The Long Short-Term Memory (LSTM) layer processes sequential data by capturing long-range dependencies. Unlike traditional RNNs, LSTMs utilize gates (forget, input, and output) to manage the flow of information, enabling the model to retain or discard information across time-steps. This is essential for tasks like text classification, where context is key.

The attention mechanism allows the model to focus on the most relevant parts of the input sequence. It assigns attention weights to different elements based on their importance, creating a context vector that enhances the model's understanding of the input.

This method selects the maximum value from each feature map, helping extract the most prominent features across the entire sequence.

It calculates the average of all values in each feature map, reducing dimensionality and preventing overfitting by limiting model complexity.

Fully connected layers combine the learned features to make predictions. The complexity of the dense layers can vary, with deeper networks requiring more data to avoid overfitting.

The output layer typically uses a sigmoid activation function for binary classification tasks, generating a probability between 0 and 1. A threshold (usually 0.5) is applied to determine the final class label (0 or 1).

Model Evaluation and Prediction:

Binary Crossentropy: For binary classification, this loss function measures the difference between predicted probabilities and actual labels, guiding the model to minimize this loss during training.

Metrics:

Accuracy: The proportion of correct predictions.

Precision, Recall, F1-Score: These metrics provide deeper insights, especially for imbalanced datasets where false positives or negatives carry significant weight.

This architecture is designed to effectively handle sequential data, like text classification tasks, by leveraging the temporal dependencies captured by LSTMs and the focus-enhancing capabilities of attention mechanisms. Regularization techniques like dropout and pooling help prevent overfitting, while the dense layers and sigmoid output layer are ideal for binary classification tasks like sarcasm detection.

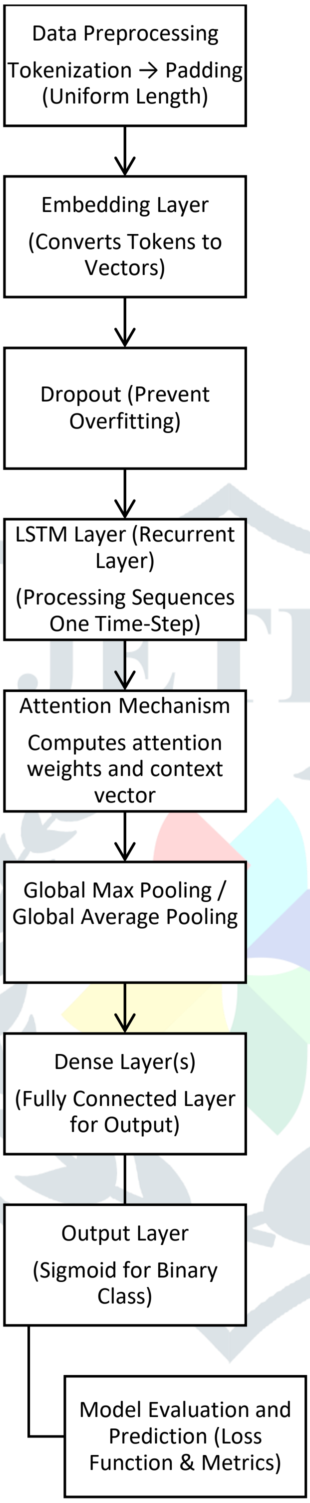

Figure 2: Flowchart of Bi-LSTM model

RoBERTa Model: Architecture, Fine-Tuning Process, and Hyperparameters

InputText

The RoBERTa model starts with raw text input, which could range from a single sentence to a longer paragraph, depending on the task. For instance, a sample input could be the tweet:

“Oh great, another Monday. Just what I needed!”

This text is processed by the model as the initial step.

Tokenization

Tokenization involves breaking down the input text into smaller components called tokens. RoBERTa uses Byte-Pair Encoding (BPE), which splits words into subwords or word pieces. This approach is effective in handling rare or out-of-vocabulary words.

For example, the sentence:

“Oh great, another Monday. Just what I needed!”

would be tokenized as:

['Oh', 'great', ',', 'another', 'Monday', '.', 'Just', 'what', 'I', 'needed', '!']

RoBERTa adds special tokens to mark the beginning ([CLS]) and the end ([SEP]) of the sequence. These tokens are essential for certain tasks, such as classification. The tokenized words are then converted into input IDs, numerical representations of each token (e.g., the token "great" might map to ID 1234).

Embedding Layer

The token IDs are passed through an embedding layer, which converts each token into a dense vector. These embeddings capture both semantic and syntactic features of the tokens. For example, the word “great” might be represented as a 768-dimensional vector (for RoBERTa-base). The embedding layer also includes positional embeddings to retain information about the order of tokens in the sequence, which is necessary since transformers don’t inherently understand token order.

Transformer Encoder

RoBERTa employs multiple transformer encoder layers (12 for the base version and 24 for the large version). Each encoder layer contains:

1. **Multi-Head Self-Attention**: This mechanism allows the model to focus on various parts of the input sequence by computing attention scores between every pair of tokens. RoBERTa uses multi-head attention to simultaneously capture different types of relationships, such as semantic and syntactic dependencies.
2. **Feed-Forward Neural Network (FFN)**: After attention, the output is passed through a feed-forward network comprising two linear transformations, with a non-linear activation function like GELU in between. This further processes the information and extracts higher-level features.

The encoder layers also incorporate residual connections and layer normalization to stabilize training and improve the flow of gradients.

Pooling (Optional)

For tasks like classification, the output of the transformer encoder can be pooled. The most common approach is to use the embedding of the [CLS] token as a summary representation of the entire input sequence. This helps in capturing the overall meaning or sentiment of the text, which is useful for tasks like sentiment analysis or sarcasm detection.

Task-Specific Head

The pooled representation or the sequence output is passed through a task-specific head, which adapts the model for the given task. For classification tasks, this head usually consists of a fully connected layer followed by a softmax activation function. The fully connected layer transforms the high-dimensional representation (e.g., 768-dimensional for RoBERTa-base) to match the number of output classes. For example, in binary classification, the output might look like:

[0.9, 0.1], indicating a 90% probability for one class (e.g., sarcastic) and a 10% probability for the other (e.g., not sarcastic).

Loss Calculation

During training, the model’s predictions are compared to the true labels using a loss function. In classification tasks, the cross-entropy loss function is typically used to compute the difference between the predicted probabilities and the actual labels. For instance, if the correct label is “sarcastic,” and the model predicts a low probability for this class, the loss function will penalize the model.

Backpropagation

The loss is used to compute gradients through backpropagation, which are then used to adjust the model’s weights. An optimizer like AdamW updates the model parameters to minimize the loss and enhance its performance over time. For example, if the model incorrectly classifies a sarcastic tweet, backpropagation will adjust the model’s weights to reduce such errors in the future.

Prediction

After training, the model can be used for predictions on unseen data. The input text undergoes the same process: tokenization → embedding → transformer encoder → pooling → task-specific head. The model produces a probability distribution over the output classes, with the class having the highest probability being selected as the final prediction. For example, given the input "Oh great, another Monday," the model might output [0.85, 0.15], indicating an 85% chance the input is sarcastic.

Training Details

Both RoBERTa and similar models are trained using batch processing, where the data is divided into smaller batches. Training typically involves several epochs, with callbacks like ReduceLROnPlateau being used to adjust the learning rate during training. This helps prevent overfitting and ensures better convergence of the model's parameters.

for Bi-LSTM).

Evaluation Metrics: Discuss the evaluation metrics (accuracy, loss, etc.) used to compare the performance of the models.

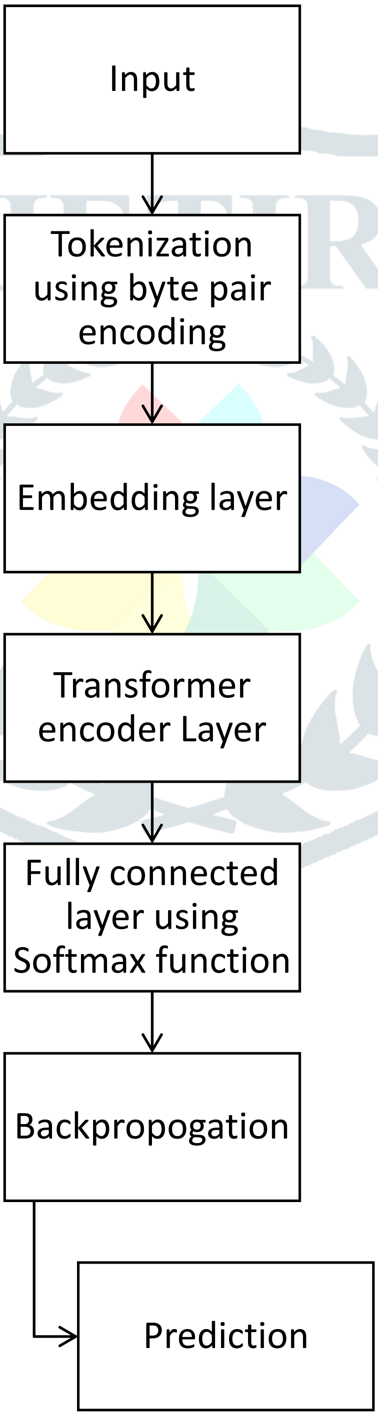

Figure 3:Flowchart of Roberta Model .

D.Difference Between the Architecture of BiLSTM and RoBERTa Models

BiLSTM Architecture:

BiLSTM is an enhancement of the Long Short-Term Memory (LSTM) network, designed to mitigate the vanishing gradient problem common in traditional Recurrent Neural Networks (RNNs). The BiLSTM architecture processes the input sequence using two LSTM layers: one processes it from left to right (forward direction), and the other from right to left (backward direction). This bidirectional approach allows BiLSTM to capture both past and future context for each token at every time step. The results from both directions are combined to create a comprehensive context-aware representation for each token. However, BiLSTM processes the data sequentially, which makes it less efficient for handling long sequences or parallel processing tasks.

RoBERTa Architecture:

RoBERTa is a transformer-based model derived from BERT, but it includes several improvements. It uses a stack of Transformer encoder layers, each incorporating multi-head self-attention and position-wise feed-forward networks. Unlike RNNs that process tokens sequentially, RoBERTa can consider all tokens in a sequence at the same time through self-attention. This allows the model to capture dependencies across tokens, irrespective of their positions in the sequence. Key differences from BERT include RoBERTa's removal of the Next Sentence Prediction (NSP) task, the use of dynamic masking during training, and its pretraining on a larger corpus with longer sequences and bigger batch sizes, resulting in improved generalization and performance.

Data Processing and Context Representation:

BiLSTM uses time-dependent memory, where the output of each token is influenced by both past and future tokens due to its bidirectional design. However, it processes the data sequentially, which makes it less efficient in capturing long-range dependencies. In contrast, RoBERTa utilizes absolute and learned positional encodings combined with self-attention mechanisms, allowing every token to interact with all other tokens in the sequence simultaneously. This leads to a richer and more effective representation of the context, particularly for modeling long-range dependencies.

Training Objectives and Strategies:

BiLSTM models are typically trained for specific tasks like part-of-speech tagging, named entity recognition, or sentiment classification using supervised learning on labeled data. These models are usually not pretrained on large datasets and require extensive domain-specific fine-tuning.

On the other hand, RoBERTa follows a pretrain-then-finetune paradigm. Initially, it is trained in an unsupervised manner with the Masked Language Modeling (MLM) objective, where parts of the input text are masked, and the model learns to predict them. Once pretrained, RoBERTa is fine-tuned on downstream tasks with relatively small amounts of labeled data, which makes it highly versatile across various applications.

Efficiency and Scalability:

BiLSTM is computationally more lightweight, making it suitable for smaller datasets and lower-resource hardware. However, its sequential nature limits its ability to scale effectively for longer sequences, leading to slower training and inference times. RoBERTa, utilizing the Transformer architecture, excels in large-scale training scenarios due to its parallelizable structure. It benefits from distributed computing resources during pretraining, which allows it to handle vast amounts of data. However, RoBERTa's high memory and computational demands typically require the use of GPUs or TPUs for efficient operation.

Performance and Generalization:

RoBERTa consistently outperforms BiLSTM in benchmark NLP tasks like GLUE, SQuAD, and sentiment analysis. Its ability to model language holistically and generate deeper, abstract representations of text enables better generalization across diverse tasks and languages.

Although BiLSTM is effective for many sequence-based tasks, its performance typically suffers in complex scenarios where long-range dependencies or nuanced semantic understanding are essential. The sequential processing limitation of BiLSTM and its dependence on training data can also restrict its performance.

4.RESULTS AND DISCUSSION

Table 4.1: Evaluation results of RoBERTa and Bi-LSTM with respect to Epochs.

EPOCHS	ROBERTA	LSTM
2	Val Loss: 0.2137, Val Acc: 0.9280, Val F1: 0.9210	Val 0.3725 Val Acc: 0.8311 Val F1 : 0.8236
4	Val Loss: 0.3603, Val Acc: 0.9285, Val F1: 0.9225	Val Loss: 0.5033 Val Acc: . 0.8390, Val F1: 0.8322
6	Val Loss: 0.4232, Val Acc: 0.9306, Val F1: 0.9241	Val Loss: 0.5844, Val Acc: 0.8330, Val F1: 0.8363
8	Val Loss: 0.4500, Val Acc: 0.9352, Val F1: 0.9300	Val Loss: 0.8072, Val Acc: 0.8430, Val F1: 0.8352
10	Val Loss: 0.4767, Val Acc: 0.9376, Val F1: 0.9326	Val 0.5096 Val Acc: 0.8470 Val F1 : 0.8403

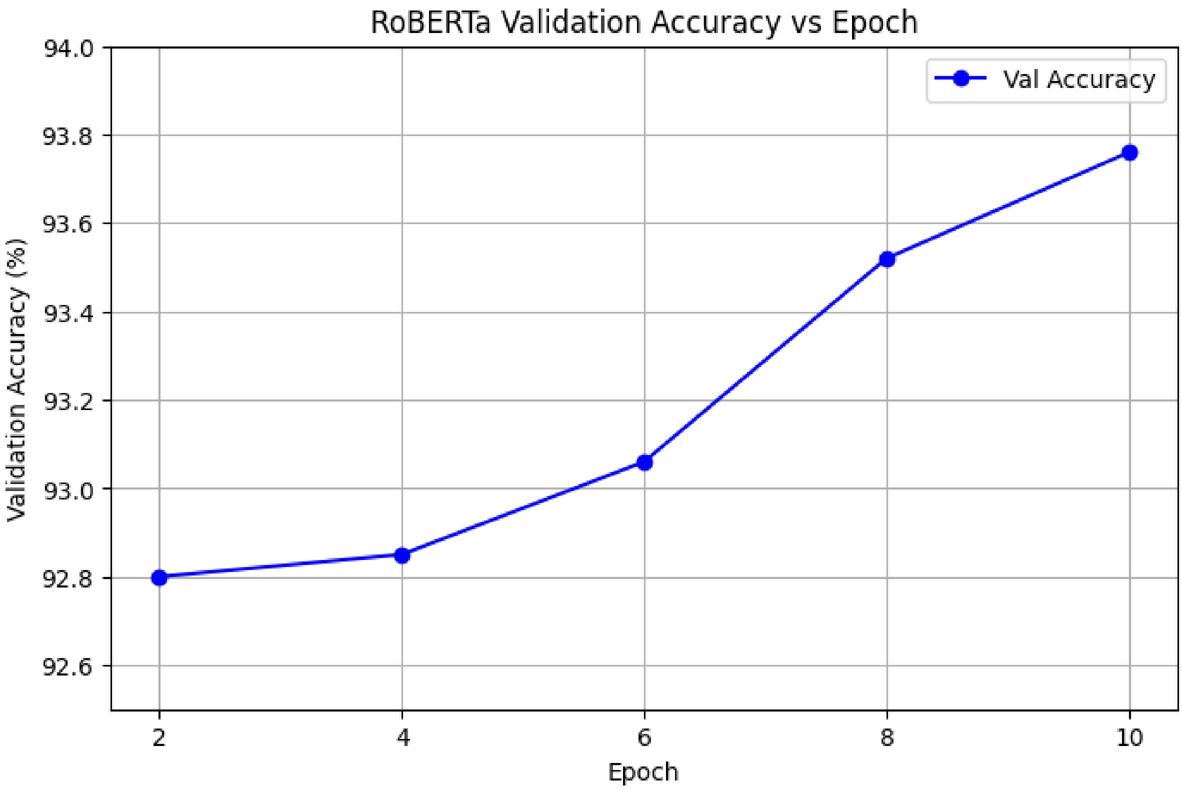

Figure 4 : graph consisting of validation accuracy vs Epoch of Roberta Model.

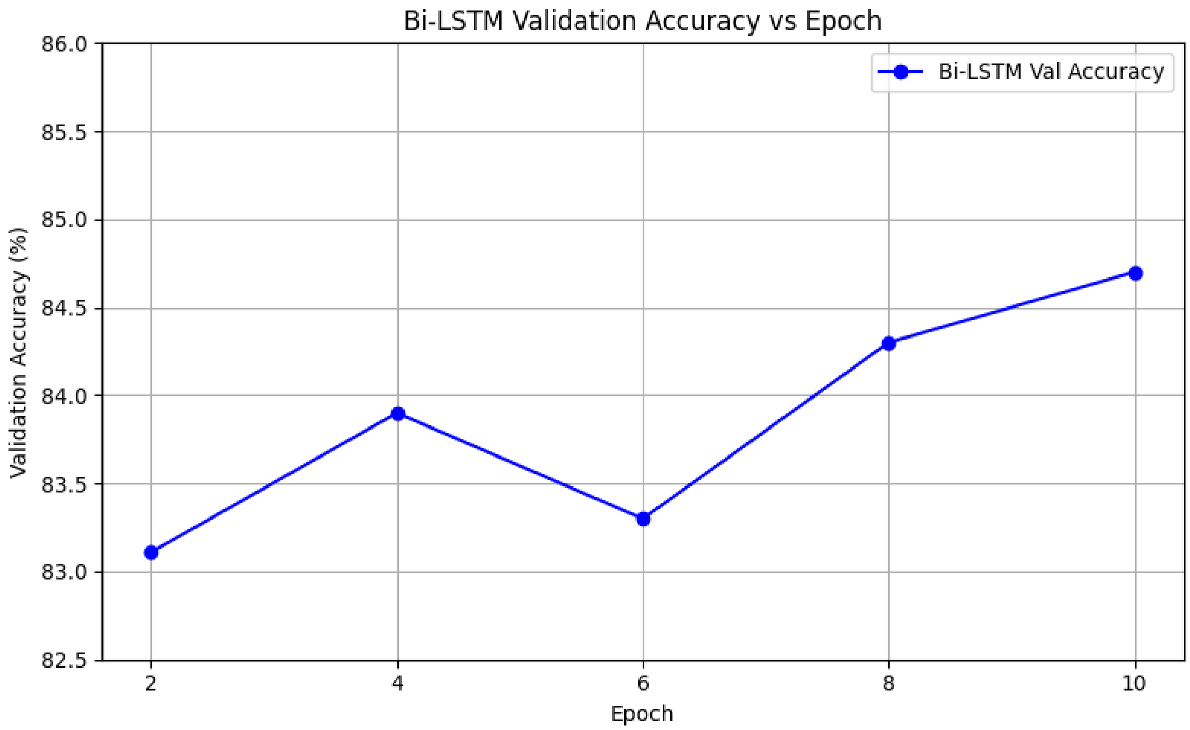

Figure 5 : graph consisting of validation accuracy vs Epoch of Bi-LSTM Model.

Performance Comparison Between RoBERTa and Bi-LSTM

This section presents a comparative analysis of the performance of two deep learning models—RoBERTa and Bi-LSTM—used for text classification. The evaluation metrics include validation accuracy, validation F1-score, and validation loss across multiple epochs (2, 4, 6, 8, and 10).

A.Quantitative Results

The RoBERTa model consistently outperformed the Bi-LSTM model across all metrics. At epoch 10, RoBERTa achieved a validation accuracy of 93.76% and a validation F1-score of 0.9326, whereas Bi-LSTM achieved a validation accuracy of 84.70% and an F1-score of 0.8403. This pattern remained consistent across all epochs ,with RoBERTa outperforming Bi-LSTM by approximately 8–10% in both accuracy and F1-score.

B. Architectural Comparison

The superior performance of RoBERTa can be attributed to its transformer-based architecture, which uses self-attention mechanisms to model global dependencies across an entire input sequence simultaneously. Unlike Bi-LSTM, which processes text in a sequential manner (both forward and backward), transformers allow for parallel processing and capture contextual relationships regardless of token position. This makes RoBERTa highly effective in understanding the semantic nuances of language.

RoBERTa gains from extensive pre-training on large-scale datasets using masked language modeling, which improves its ability to generalize across various downstream tasks. On the other hand, Bi-LSTM, while adept at processing sequential data, often faces challenges with long-range dependencies due to its sequential processing nature and limited capacity.

C. Summary of Findings

The comparative results highlight that RoBERTa outperforms Bi-LSTM in terms of accuracy and robustness for natural language processing tasks. Table 5.2 provides a summary of the validation metrics across epochs:

Table 4.2: Performance Comparison of RoBERTa and Bi-LSTM

Epoch	RoBERTa Val Accuracy	RoBERTa Val F1	Bi-LSTM Val Accuracy	Bi-LSTM Val F1
2	92.80%	0.9210	83.11%	0.8236
4	92.85%	0.9225	83.90%	0.8322
6	93.06%	0.9241	83.30%	0.8363
8	93.52%	0.9300	84.30%	0.8352
10	93.76%	0.9326	84.70%	0.8403

These results highlight RoBERTa's suitability for tasks requiring a deep understanding of context and semantics in textual data. The model's architectural advancements make it a more powerful alternative to traditional RNN-based methods like Bi-LSTM.

D. Use Cases and Advantages

BiLSTM Model

NamedEntityRecognition(NER)

Named Entity Recognition is a key task in natural language processing that focuses on identifying and categorizing entities like people, organizations, locations, dates, and numerical expressions within text. BiLSTM models are well-suited for this task because of their bidirectional design, which helps capture contextual information from both the preceding and following words in a sentence. This ability is crucial for accurately identifying entities, especially when their meaning depends on the surrounding context.

SpeechRecognition

In speech-to-text tasks, BiLSTM models excel by capturing long-term dependencies in spoken language. Speech often contains patterns or cues that span over several time steps, and BiLSTM's ability to process information from both directions enhances its capability to recognize and interpret these patterns, even in noisy or acoustically challenging environments.

Part-of-SpeechTagging

Part-of-speech tagging involves assigning grammatical categories (such as noun, verb, or adjective) to words in a sentence. BiLSTM performs well in this area because it can process contextual information from both directions. The correct tag for a word often depends on the words that follow it in the sentence, and the bidirectional architecture helps improve accuracy by providing a more complete understanding of the syntactic structure.

Sentiment Analysis for Short Texts

BiLSTM models are particularly effective for sentiment analysis of short texts, such as tweets, product reviews, or brief comments. Since short texts often contain sentiment indicators at any point in the sequence, including the end, BiLSTM's ability to analyze both past and future context enables it to detect subtle shifts in sentiment, resulting in more accurate and nuanced sentiment classification.

RoBERTa Model

Question Answering (QA)

In Question Answering tasks, where the system needs to provide answers based on a given passage, RoBERTa excels. Its transformer-based architecture enables it to attend to both the question and the passage simultaneously, allowing the model to pinpoint and focus on the most relevant segments. This feature greatly enhances the accuracy and efficiency of extracting precise answers from the context.

MachineTranslation

RoBERTa is highly effective for machine translation tasks due to its ability to process entire sequences of text at once. This holistic view of the input allows it to better understand relationships and dependencies across different parts of a sentence, which is particularly beneficial for translating languages with complex grammatical structures or idiomatic expressions.

TextSummarization

In text summarization tasks, where the goal is to produce concise and coherent summaries of long documents, RoBERTa's self-attention mechanism helps identify key information throughout the entire document. This enables the model to generate meaningful summaries that highlight the most important points. Its capacity to understand the global context ensures that the generated summaries are both relevant and coherent.

Sentiment Analysis for Longer Texts

RoBERTa also performs well in sentiment analysis of longer texts, such as comprehensive reviews or social media posts. Its transformer architecture can handle extended input sequences, preserving important contextual cues that span across multiple sentences or paragraphs. This makes it particularly adept at analyzing sentiment when it is shaped by the overall narrative or structure of the document.

5. Conclusion

This study compares the performance of two distinct neural network architectures—RoBERTa (a transformer-based model) and Bi-LSTM (a recurrent neural network)—for sarcasm detection in textual data. Evaluation metrics such as validation accuracy, F1-score, and loss were tracked across multiple epochs. The results indicated that RoBERTa significantly outperformed Bi-LSTM in all areas. At its peak performance (epoch 10), RoBERTa achieved a validation accuracy of 93.76% and a validation F1-score of 0.9326, while Bi-LSTM attained a validation accuracy of 84.70% and an F1-score of 0.8403. These findings emphasize the superior ability of transformer-based models to capture the nuanced and context-dependent nature of sarcasm in natural language.

Despite the promising outcomes, the study has several limitations. Firstly, the dataset used was limited in both size and domain, which could impact the model's generalizability to different forms or contexts of sarcasm. Secondly, transformer-based models like RoBERTa are computationally demanding, requiring substantial hardware resources and longer training times compared to Bi-LSTM. While RoBERTa effectively captures deep contextual understanding, it may still encounter difficulties with highly implicit or culturally specific forms of sarcasm.

The findings highlight the importance of choosing the right model for specific natural language processing tasks. Transformer-based models like RoBERTa are particularly effective for sarcasm detection due to their ability to understand complex dependencies and contextual information. As the field of NLP progresses, selecting the appropriate architecture for each task will continue to be crucial in developing intelligent and reliable language systems.

6. Future Scope

This study opens up several potential avenues for further exploration and development in the field of sarcasm detection using deep learning models. While RoBERTa has demonstrated superior performance over Bi-LSTM in this study, there remains significant potential for improvement. One promising direction is to fine-tune RoBERTa on larger, more diverse, and domain-specific sarcasm datasets. This approach could enhance the model's ability to generalize across various contexts, including political, humorous, or culturally specific discourse.

Another avenue to explore is the development of hybrid models that combine the sequential strengths of Bi-LSTM with the contextual richness of transformer models like RoBERTa. Such hybrid models could leverage both local syntactic patterns and global contextual understanding, which might further improve accuracy and F1-scores.

Additionally, future research could extend beyond text-based sarcasm detection to multi-modal detection, integrating visual and auditory cues such as tone of voice and facial expressions. This would enhance the robustness of models for real-world applications, including video content analysis, virtual assistant interactions, and social media video content filtering.

From an application perspective, sarcasm detection systems could be implemented in social media monitoring tools, customer service platforms, and mental health analysis systems, where understanding nuanced sentiments could improve service and support. These systems could also be integrated into news aggregators or content moderation tools to detect sarcastic misinformation or biased commentary.

Lastly, future research should focus on improving model efficiency and reducing computational demands. Approaches like knowledge distillation, quantization, or pruning could make transformer models like RoBERTa more accessible for deployment in low-resource environments, enabling real-time sarcasm detection on mobile or edge devices.

REFERENCES

[1] Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. 2019. Towards Multimodal Sarcasm Detection (An _Obviously_ Perfect Paper). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4619–4629, Florence, Italy. Association for Computational Linguistics.

[2] Javed, Tayyaba & Nauman, Asif & Zahid, Rushna. (2024). BERT Model Adoption for Sarcasm Detection on Twitter Data. VFAST Transactions on Software Engineering. 12. 177. 10.21015/vtse.v12i3.1908.

[3] Abaskohi, Amirhossein & Rasouli, Arash & Zeraati, Tanin & Bahrak, Behnam. (2022). UTNLP at SemEval-2022 Task 6: A Comparative Analysis of Sarcasm Detection using generative-based and mutation-based data augmentation. 10.18653/v1/2022.semeval-1.135.

[4] Hongliang Pan, Zheng Lin, Peng Fu, Yatao Qi, and Weiping Wang. 2020. Modeling Intra and Inter-modality Incongruity for Multi-Modal Sarcasm Detection. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1383–1392, Online. Association for Computational Linguistics.

[5] S. Khotijah, J. Tirtawangsa, and A. A. Suryani, "Using LSTM for context-based approach of sarcasm detection in Twitter," in Proc. 11th Int. Conf. Adv. Inf. Technol., 2020, pp. 1–7.[

[6] Chy, M.S.R., Chy, M.S.R., Mahin, M.R.H., Rahman, M.M., Hossain, M.S., Rasel, A.A. (2023). Sarcasm Detection in News Headlines Using Evidential Deep Learning-Based LSTM and GRU. In: Lu, H., Blumenstein, M., Cho, SB., Liu, CL., Yagi, Y., Kamiya, T. (eds) Pattern Recognition. ACPR 2023. Lecture Notes in Computer Science, vol 14406. Springer, Cham. https://doi.org/10.1007/978-3-031-47634-1_15

[7] Olaniyan, D., Ogundokun, R. O., Bernard, O. P., Olaniyan, J., Maskeliūnas, R., & Akande, H. B. (2023). Utilizing an Attention-Based LSTM Model for Detecting Sarcasm and Irony in Social Media. *Computers*, *12*(11), 231. https://doi.org/10.3390/computers12110231

[8] Kumar, A., Narapareddy, V.T., Veerubhotla, A., Malapati, A., & Neti, L.B. (2020). Sarcasm Detection Using Multi-Head Attention Based Bidirectional LSTM. *IEEE Access, 8*, 6388-6397.

[9] H. Liu and L. Xie, "Research on Sarcasm Detection of News Headlines Based on Bert-LSTM," *2021 IEEE International Conference on Emergency Science and Information Technology (ICESIT)*, Chongqing, China, 2021, pp. 89-92, doi: 10.1109/ICESIT53460.2021.9696851.

[10] J. Dai, "A BERT-Based with Fuzzy logic Sentimental Classifier for Sarcasm Detection," *2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE)*, Shanghai, China, 2024, pp. 1275-1280, doi: 10.1109/ICAACE61206.2024.10548550.

[11] Bhat and A. Chauhan, "A Deep Learning based approach for MultiModal Sarcasm Detection," *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Greater Noida, India, 2022, pp. 2523-2528, doi:10.1109/ICAC3N56670.2022.10074506.

[12] Bhat and G. N. Jha, "Sarcasm Detection of Textual Data on Online SocialMedia: A Review," *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, Greater Noida, India, 2022, pp. 1981-1985, doi: 10.1109/ICACITE53722.2022.9823869.[

[13] K. Bari, "Sarcasm Detection of Newspaper Headlines Using LSTM-RNN," *2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Greater Noida, India, 2023, pp. 604-608, doi: 10.1109/ICAC3N60023.2023.10541797

[14] Sharma, A. U. Pandey and A. Gupta, "Sarcasm Detection on News Headline Dataset Using Language Models," *2023 3rd International Conference on Intelligent Technologies (CONIT)*, Hubli, India, 2023, pp. 1-6, doi: 10.1109/CONIT59222.2023.10205839]

[15] K. Srivastava and R. Srivastava, "Advancements in Sarcasm Detection: A Review of Accurate Models for Short Text Data," *2024 4th International Conference on Advancement in Electronics & Communication Engineering (AECE)*, GHAZIABAD, India, 2024, pp. 1002-1006, doi: 10.1109/AECE62803.2024.10911701

[16] R. Kumar and S. Sinha, "Comprehensive Sarcasm Detection using Classification Models and Neural Networks," *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 2022, pp. 1309-1314, doi: 10.1109/ICACCS54159.2022.9785305

[17] S. R, P. R, S. T and T. K, "LSTM with Ensemble ML Algorithms for Sentimental Analysis," *2024 5th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India, 2024, pp. 1234-1237, doi: 10.1109/ICESC60852.2024.10689886

[18] Pandey and D. K. Vishwakarma, "Multimodal Sarcasm Detection (MSD) in Videos using Deep Learning Models," *2023 International Conference in Advances in Power, Signal, and Information Technology (APSIT)*, Bhubaneswar, India, 2023, pp. 811-814, doi: 10.1109/APSIT58554.2023.10201731

[19] Kitanovski, M. Toshevska and G. Mirceva, "DistilBERT and RoBERTa Models for Identification of Fake News," *2023 46th MIPRO ICT and Electronics Convention (MIPRO)*, Opatija, Croatia, 2023, pp. 1102-1106, doi: 10.23919/MIPRO57284.2023.10159740.

[20] Băroiu and Ş. Trăuşan-Matu, "How capable are state-of-the-art language models to cope with sarcasm?," *2023 24th International Conference on Control Systems and Computer Science (CSCS)*, Bucharest, Romania, 2023, pp. 399-402, doi: 10.1109/CSCS59211.2023.00069

[21] N. Rayvanth, S. S. S, V. K. R. Challa, V. Sharma and M. Venugopalan, "Exploring Sarcasm Detection: Leveraging Neural Network Models with BERT Embeddings," *2024 4th International Conference on Intelligent Technologies (CONIT)*, Bangalore, India, 2024, pp. 1-6, doi: 10.1109/CONIT61985.2024.10626842

[22] I. Eke, A. A. Norman and L. Shuib, "Context-Based Feature Technique for Sarcasm Identification in Benchmark Datasets Using Deep Learning and BERT Model," in *IEEE Access*, vol. 9, pp. 48501-48518, 2021, doi: 10.1109/ACCESS.2021.3068323