Journal on
Artificial Intelligence

Tech Science Press

Doi:10.32604/jai.2025.072531

ARTICLE

Why Transformers Outperform LSTMs: A Comparative Study on Sarcasm Detection

Palak Bari, Gurnur Bedi, Khushi Joshi and Anupama Jawale*

Department of Information Technology, Narsee Monjee College of Commerce and Economics, Mumbai, 400056, Maharashtra, India

*Corresponding Author: Anupama Jawale. Email: anupama.jawale@nmcce.ac.in

Received: 29 August 2025; Accepted: 27 October 2025; Published: 14 November 2025

ABSTRACT: This study investigates sarcasm detection in text using a dataset of 8095 sentences compiled from MUStARD and HuggingFace repositories, balanced across sarcastic and non-sarcastic classes. A sequential baseline model (LSTM) is compared with transformer-based models (RoBERTa and XLNet), integrated with attention mechanisms. Transformers were chosen for their proven ability to capture long-range contextual dependencies, whereas LSTM serves as a traditional benchmark for sequential modeling. Experimental results show that RoBERTa achieves 0.87 accuracy, XLNet 0.83, and LSTM 0.52. These findings confirm that transformer architectures significantly outperform recurrent models in sarcasm detection. Future work will incorporate multimodal features and error analysis to further improve robustness.

KEYWORDS: Attention mechanism; LSTM; natural language processing; sarcasm detection; sentiment analysis; transformer models; RoBERTa; XLNet

1 Introduction

Sarcasm [1] is a subtle form of communication in which literal and intended meanings diverge, making it particularly difficult for NLP systems to detect. Unlike sentiment expressed directly, sarcastic statements rely heavily on contextual cues and often resemble non-sarcastic text. This poses challenges for sentiment analysis in social media, reviews, and dialogue systems. Moreover, sarcasm may be expressed by visual cues, such as facial expressions and gestures. Prosodic qualities function as aural markers of sarcasm [2]. A proficient speaker can convey sarcasm with ease, and it is essential to develop both visual and auditory recognition of this form of expression. A straightforward assertion such as "Funny is all I have" may be understood as sarcastic based on the relevant vocal or visual indicators present. Recent work has applied both traditional models (e.g., SVM, LSTM) and transformer-based architectures (e.g., BERT, DeBERTa, ChatGPT) to sarcasm detection, with transformers generally outperforming recurrent models. However, systematic comparisons across different architectures under controlled settings remain limited [3]. Emotions significantly influence the identification of sarcasm, as the essence of this phenomenon is rooted in human comprehension. Determining the level of human intelligence presents challenges for devices; however, successful training and testing can enhance their capabilities, making them more effective and advantageous.

In this study, a corpus of 8095 sentences is constructed by combining the MUStARD dataset with sarcasm data from HuggingFace repositories. Three representative models: LSTM (baseline sequential model), RoBERTa, and XLNet (transformer-based models), are compared, each with and without attention mechanisms [4].

Developing an NLP system typically requires the manual establishment of rules and linguistic resources, a process that can be both time-consuming and specialized. In contrast, Large Language Models (LLMs) [5] utilize automated training on extensive datasets, necessitating considerable computational resources and expertise in deep learning techniques. This approach significantly accelerates the process compared to traditional NLP methods [6]. LLMs provide a distinct advantage in this regard and have been incorporated into this research study.

2 Contributions

Our contributions are fourfold: (i) development of a consolidated sarcasm detection dataset; (ii) comparative evaluation of sequential and transformer architectures; (iii) inclusion of error analysis to highlight cases where transformers outperform LSTMs; and (iv) statistical significance testing of observed performance improvements.

3 Related Work

Sarcasm detection has been a growing area of interest due to its complex nature, particularly in the context of social media, where brief, informal, and multimodal expressions dominate. Prior studies have approached sarcasm detection through various traditional, deep learning, and multimodal techniques. The related work in this domain can be summarized as follows.

i. Text-based approach

Early sarcasm detection relied on traditional ML models such as SVM and logistic regression, often using lexical, pragmatic, and sentiment features. Traditional machine learning techniques also remain relevant. Study [7] found SVM to be the most effective model for sarcasm detection on Twitter, with performance enhanced by lexical, pragmatic, and part-of-speech features. A hybrid CNN-SVM approach further improved performance by incorporating both lexical and personal information. A two-phase sarcasm detection framework was proposed in [8], where Phase 1 utilizes models such as BERT and fastText to determine whether a statement belongs to humorous or non-humorous categories. Ensemble methods including Random Forest, Support Vector Machines (SVM), and Logistic Regression were employed to improve detection efficiency. In Phase 2, deep learning models like TD-LSTM and TC-LSTM analyzed sarcasm within humorous phrases by focusing on sentence context, word-level semantics, and sarcasm target recognition.

The research [9] utilized a variety of lexical, pragmatic, and sentiment-based features and found that Bi-directional LSTM outperformed CNN and LSTM models, with accuracies of 86.32% and 82.91%, respectively. Deep learning methods including CNNs, LSTMs, and Bi-LSTMs later improved performance by modeling sequential context. Significant progress has also been made with hybrid and optimized models. The DLNLP-SA model [10], which uses N-gram feature extraction, MHSA-GRU, and Metaheuristic Feature Optimization (MFO), achieved an outstanding 97.61% accuracy and over 94% F1 score. The model was especially effective on Twitter and dialogue datasets and focused on negative emotional cues to detect sarcasm. Furthermore, the study in [11] experimented with various deep learning models including CNN, LSTM, and GRU, combined with pre-trained word embeddings such as Word2Vec, GloVe, and fastText. The best performing models used an 80:20 train-validation split, showing a gradual accuracy improvement: CNN < LSTM < Bi-LSTM, with Bi-LSTM delivering the highest accuracy.

ii. Transformer-based models

Together, these studies underscore the growing sophistication of sarcasm detection models, particularly those that leverage deep learning, multimodal signals, context-aware attention, and hybrid architectures.

However, the field still faces key challenges such as language diversity, implicit meaning detection, and fine-grained context modeling, indicating that further innovation is needed in both model design and dataset development. Multimodal approaches have also demonstrated significant promise. A study in [12] used RoBERTa and RCNN to identify snark and irony, highlighting the utility of transformer-based architectures.

Several innovations have been proposed to enhance multimodal sarcasm detection. The MMOE (Multimodal Mixtures of Experts) framework was developed to better handle multiple input modalities, while MOSES (MOdelling Stand rESponse) [13] integrated deep neural networks with natural language explanations to identify sarcasm and related emotional content in dialogues. In [14], image-text incongruity was leveraged using ResNet and co-attention mechanisms, achieving a 6.14% increase in F1 score over baseline models. Transformer-based and aspect-based sentiment analysis methods have also been shown effective for sarcasm detection [13–17]. These approaches address the challenges posed by implicit signals and contextual nuances. Notably, MOSES, CLFA, and ChatGPT have all contributed significantly to this domain, advancing affective computing and sarcasm understanding.

iii. Multimodal approaches

Sarcasm is not only textual but often conveyed through audio and visual signals. The MUStARD dataset [10] enabled multimodal sarcasm detection, where models incorporating text, audio, and video outperform text-only baselines by large margins, such as Ref. [18] emphasized that multimodal models, especially those using both audio and video inputs, outperform unimodal ones in sarcasm identification tasks. The MUStARD dataset [19] provides sarcastic audiovisual sentences and has been pivotal in benchmarking performance. The study noted a 12.9% improvement in F1 score when using multimodal models compared to text-only baselines. In another study, Ref. [20] proposed a BiGRU framework using Bayesian priors to model sarcasm in diverse contexts, outperforming models like CASCADE. In [21], a CNN-based sarcasm detector was enhanced with user embeddings, reflecting individual user behavior patterns.

iv. Hybrid and optimized frameworks.

Language diversity introduces further complexity. According to [22], while English corpora are abundant, sarcasm recognition in regional or code-switched languages presents significant challenges. A novel context-sensitive sentiment analysis method, RO-TGANN (Remora-Optimized Twofold Gated Attention Neural Network), was proposed in [23], focusing on accurate sentiment and context modeling.

Psycholinguistic factors have also been considered. Study [24] explored how sentiment, mood, and personality traits influence sarcasm recognition by deep CNNs. In multilingual and code-switching scenarios, Ref. [25] demonstrated that Hierarchical Attention Networks (HAN) improved Hindi-English sarcasm detection by 4.7% in F1 score, highlighting the value of modeling language-switch patterns.

Visual modalities have also been explored further. For instance, Ref. [26] incorporated contextual linkages and utterance sequences in video-based sentiment analysis using LSTM-based models. The model in [27] introduced Image-Text Contrastive (ITC) and Image-Text Matching (ITM) auxiliary tasks to enhance multimodal performance. Meanwhile, the Hyphen Model [28] integrated hyperbolic Fourier co-attention and hierarchical graph structures to model public sentiment from source posts and comments.

Despite significant progress, two limitations remain: (i) insufficient systematic comparisons between traditional recurrent models (e.g., LSTM) and transformers under controlled conditions, and (ii) limited attention to reproducibility, error analysis, and statistical validation. This study addresses these gaps by evaluating LSTM, RoBERTa, and XLNet on a unified sarcasm corpus, providing both quantitative and qualitative comparisons.

4 Methodology

A dataset of 8095 sentences is compiled by combining MUStARD with a publicly available sarcasm corpus from HuggingFace. The dataset is approximately balanced between sarcastic and non-sarcastic classes. Text was preprocessed by lowercasing, removing special symbols, normalizing punctuation, and handling emojis. Transformers (RoBERTa, XLNet) used their respective HuggingFace tokenizers, while LSTM was trained on sequences padded and indexed with pre-trained GloVe embeddings. The dataset is split into 80% training, and 20% testing sets, stratified by class distribution to maintain balance.

In this research study, three representative architectures: (i) LSTM, a sequential baseline for text classification; (ii) RoBERTa, a transformer optimized for robust pre-training; and (iii) XLNet, a permutation-based transformer that captures bidirectional context are compared. Attention layers were applied to enhance token-level importance weighting.

For LSTM, Adam optimizer is used with a learning rate of 1e−3, batch size of 32, dropout rate of 0.5, and trained for 15 epochs. For transformers, AdamW is used with learning rate 2e−5, batch size of 16, and fine-tuned for 5 epochs. Cross-entropy loss was used for all models. Evaluation metrics included accuracy, precision, recall, and F1 score.

This flowchart in Fig. 1 represents a machine learning pipeline that incorporates an attention mechanism and multiple deep learning models to optimize decision-making. The process starts with the Start Node, initializing the system. The Attention Mechanism enhances model performance by focusing on important input data, improving accuracy in NLP tasks.

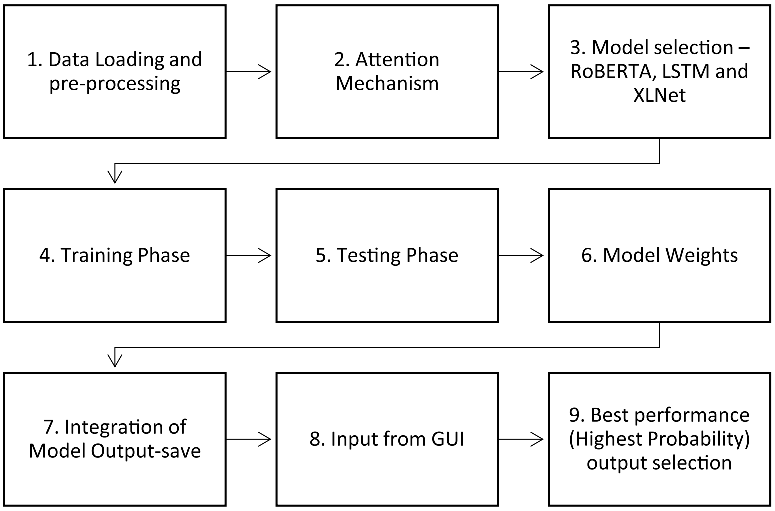

1. Data Loading and pre-processing
2. Attention Mechanism
3. Model selection – RoBERTA, LSTM and XLNet
4. Training Phase
5. Testing Phase
6. Model Weights
7. Integration of Model Output-save
8. Input from GUI
9. Best performance (Highest Probability) output selection

Figure 1: Proposed methodology

In the Model Selection phase, the system chooses between RoBERTa, LSTM, and XLNet based on the highest probability of better choice.

RoBERTa is a transformer-based model optimized for NLP, LSTM is a recurrent neural network (RNN) designed for sequential data processing, and XLNet improves upon BERT using autoregressive training. Once a model is selected, it moves to the Training Phase, where it learns patterns from labeled data, followed by the Testing Phase, where its performance is evaluated using metrics like accuracy and F1-score.

Attention Mechanism

Attention is a technique that allows neural networks to focus on important parts of the input sequence when making predictions. It is widely used in transformers, machine translation, NLP, and vision models. Instead of treating all input words equally, attention assigns different weights to different words based on their relevance to the current output, as described in Algorithm 1.

Algorithm 1: Attention calculation

```
1. Compute Attention score
      $\boldsymbol{Score}\,(\boldsymbol{Q}, \boldsymbol{K}) = \boldsymbol{QK^T}$
        Query ($Q$) represents the current word attending from
        Key ($K$) represents all words in the input sequence
        Value ($V$) is the actual representation of words after attention is applied
2. Apply Softmax to Normalize Weights
      $\alpha = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)$
        $d_k$ is the dimension of the key vectors to prevent large values
3. Multiply Attention Weights with Value Vectors
      $Attention\,(Q, K, V) = \alpha V$
        Query ($Q$) represents the current word attending from
        Key ($K$) represents all words in the input sequence
        Value ($V$) is the actual representation of words after attention is applied
```

RoBERTa

RoBERTa stands for Robustly Optimized BERT Pretraining Approach, a transformer-based model for natural language understanding which is an optimized and scaled up version of BERT. The foundation is BERT and is designed to improve performance by using hyperparameters and pretraining processes. Algorithm 2 describes working of RoBERTa, as follows.

Algorithm 2: RoBERTa implementation

```
1. Self-Attention formula
      $Attention(Q_i, K_j) = \frac{Q_i K_j^T}{\sqrt{dk}}$
      $Q_i$ is the query vector for token $i$
      $K_j$ is the key vector for token $j$
      $dk$ is the dimension of the key vector (for scaling)
2. Transformer Encoded Layer
      $AttentionOutput_i = \sum_j softmax(Q_i K_j^T / \sqrt{d_k}) V_j$
      $LayerOutput_i = FFN\,(LayerNorm\,(Attention\ Output_i))$
        FFN is a feed-forward network and LayerNorm is layer normalization
3. Loss Function for Pre-training
      $MLM = -\sum_{logP(x_i|context\ of\ x)} logP\,(x_i \mid context\ of\ x)$
        $x_i$ is the masked token at position $i$
        $P(x_i \mid context\ of\ x)$ is the model's predicted probability distribution for the
          masked token
        Masked Language Modeling (MLM)
```

LSTM

LSTM has an attention mechanism which helps to improve performance, especially the one with sequential input data, natural language processing or time-series forecasting. This combination helps the model's ability to handle long-range dependencies and select important information from sequence, making it suitable for machine translation, text classification, and time series forecasting. Algorithm 3 describes LSTM gate updates and attention algorithm used in this research.

Algorithm 3: LSTM implementation

```
1.    LSTM Gate Updates
      Input gate: $i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i)$
      Forget gate: $f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f)$
      Output gate: $o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o)$
      Cell state: $C_t = f_t.C_{t-1} + i_t.tanh(W_c.[h_{t-1}, x_t] + b_c)$
      Hidden state: $h_t = o_t.tanh(C_t)$
      $W_i$, $W_f$, $W_o$, $W_c$ are the weight matrices
      $b_i$, $b_f$, $b_o$, $b_c$ are the biases
      $\sigma$ is the sigmoid activation function
      tanh f() is the hyperbolic tangent activation function
      $x_t$ is the input at time step $t$
      $h_{t-1}$ is the previous hidden state
2.    Attention Mechanism
      Attention Score: $e_t = v^T.tanh(W_h.h_t + b_h)$
      Attention Weight: $\alpha_t = \frac{exp(e_t)}{\sum_{t=1}^{T} exp(e_t)}$
      Context Vector: $c = \sum_{t=1}^{T} \alpha_t h_t$
      v is a learnable vector
      $W_h$ is a learnable weight matrix
      $b_h$ is a biased term
3.    Final Prediction
   $y = softmax\left(W_y.c + b_y\right)$
      $W_y$ is the weight matrix
      $b_y$ is the bias term
      softmax converts the final logits into probabilities for classification tasks
```

XLNet

XLNet is a transformer model which combines the best of two worlds: built on strengths of autoregressive models (GPT) and autoencoding models (BERT). It is designed to overcome limitations of BERT leading to better performance on NLP tasks. It uses the Transformer-XL architecture (an improvement on the original Transformer) that allows the model to better handle long-range dependencies and capture more contextual information (As presented in Algorithm 4).

Algorithm 4: XLNet implementation

```
1.    Permutation Language Modeling Objective
      $P(x_i|x_{\sigma(1)}, x_{\sigma(2)}, ..., x_{\sigma(i-1)} = softmax(W_h h_i)$
        $h_i$ is the hidden state corresponding to token $x_i$ and $W_h$ is a weighted matrix
2.    Final Output and Softmax
      $P(x_i = t|context) = \frac{exp(W_h.h_i)}{\Sigma_{t^1} exp(W_h.h_i^{t^1})}$
        $h_i$ is the hidden state of the model $x_i$
        The softmax function normalizes the predicted logits across all possible tokens
3.    Final Output and Softmax
      $L = -\sum_{i=1}^{T} logP\left(x_{\sigma(1)}, x_{\sigma(2)}, ..., x_{\sigma(i-1)}\right)$
        $P$ is the set of all possible permutations of the input sequence
        $T$ is the length of the sequence
        The objective is to maximize the likelihood of predicting each token in the sequence correctly,
        based on the context
```

Next, the trained model generates Model Weights, which store learned information. If multiple models are used, their outputs are combined in the Integration of Model Outputs step using ensemble techniques for better accuracy. The system then accepts Input from UI, allowing real-time user interaction. The model processes the input and moves to Generating the Best Output, selecting the most relevant prediction. Finally, the Final Decision step delivers the optimized result, which could be a classification, recommendation, or another actionable output.

This pipeline efficiently integrates attention mechanisms, multiple deep learning models, and output fusion techniques to enhance accuracy and reliability in NLP and machine learning-based decision-making. Experimental setup for this framework is explained in next section.

5 Experimental Setup

Dataset

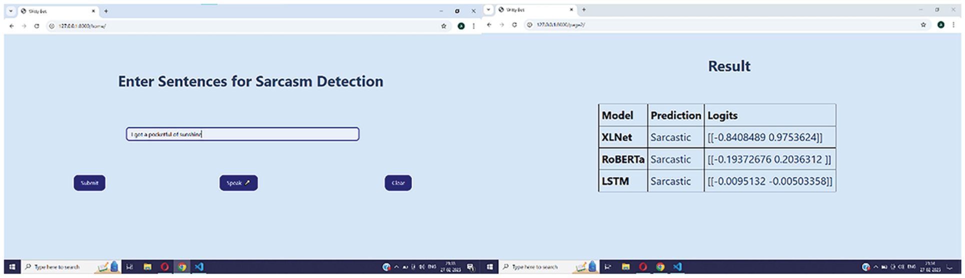
Initially the models were trained using a dataset consisting of 3648 rows and further evaluating the model by using the test dataset. Later to improve the accuracy in the models, a new dataset was created by combining the old dataset with a dataset downloaded from a repository uploaded on the hugging face, an open-source platform.

The final dataset which has been used to train all the three models (XLNET, RoBERTa and LSTM) consists of a total 8095 rows. The dataset has been split into two, train dataset and test dataset with the ratio of 80:20.

User Interface

A simple user interface is also designed using HTML/CSS and Django framework. Django is used as the web framework to manage the backend logic, routing, and rendering of web pages. It handles the HTTP requests, processes the input entered by the user and returns responses to the client accordingly. Pre-trained models are called through this interface for sarcasm detection of the input provided by user (Fig. 2).

Manual test cases have been designed to test the performance and accuracy of these models. Description of the same is provided in section below.



Enter Sentences for Sarcasm Detection

I got a pocketful of sunshine

Result

Model	Prediction	Logits
XLNet	Sarcastic	[[-0.8408489 0.9753624]]
RoBERTa	Sarcastic	[[-0.19372676 0.2036312]]
LSTM	Sarcastic	[[-0.0095132 -0.00503358]]

Figure 2: Interface design

6 Results and Discussion

Results of experimental work are presented in Table 1 as follows. Models were trained with the configurations described in Section 3. Each experiment was repeated five times with different random seeds, and results are reported as mean ± standard deviation.

Table 1: Performance comparison of different models

Model	Accuracy	Precision	Recall	F1 Score
LSTM	0.525 ± 0.012	0.525 ± 0.015	1.00 ± 0.000	0.689 ± 0.008
LSTM + Attention	0.530 ± 0.010	0.280 ± 0.013	0.530 ± 0.011	0.689 ± 0.007
RoBERTa	0.874 ± 0.006	0.852 ± 0.007	0.920 ± 0.005	0.885 ± 0.005
RoBERTa + Attention	0.871 ± 0.007	0.880 ± 0.006	0.870 ± 0.007	0.870 ± 0.005
XLNet	0.832 ± 0.009	0.850 ± 0.010	0.802 ± 0.008	0.903 ± 0.007
XLNet + Attention	0.986 ± 0.004	0.986 ± 0.004	0.986 ± 0.003	0.985 ± 0.004

The results reveal a clear performance disparity between transformer-based architectures (XLNet, RoBERTa) and the sequential baseline (LSTM) in sarcasm detection. Transformers exhibit a superior capacity to capture long-range contextual dependencies and semantic nuances, which are critical for identifying sarcastic intent. In contrast, the LSTM model demonstrates a sequential bias, processing input linearly and often missing implicit cues of irony. The notably low precision score (0.28) for LSTM + Attention indicates model overfitting to superficial lexical correlations rather than contextual semantics. Conversely, the self-attention mechanisms in XLNet and RoBERTa effectively encode inter-token dependencies, enhancing recognition of tonal and contextual shifts. Attention integration yielded a substantial gain for XLNet (F1: 0.903 → 0.985) but produced marginal or adverse effects in RoBERTa and LSTM, suggesting that XLNet's permutation-based objective inherently benefits from attention refinement over contextually salient tokens.

7 Conclusion

This research successfully developed a sarcasm detection system, using a hybrid model integrating LSTM networks, RoBERTa, and XLNet, enhanced with an attention mechanism. The primary goal is to improve sentiment analysis by accurately identifying sarcasm in text, a significant challenge for NLP. The hybrid approach leverages the strengths of each model: RoBERTa excels in contextual understanding, XLNet captures bidirectional context, and LSTM models sequential data, all refined by the attention mechanism. Experimental results showed RoBERTa achieving the highest accuracy (0.87), followed by XLNet (0.83), and LSTM (0.52), highlighting RoBERTa's superior performance in this application, before integration of attention mechanism. However, this study confirms that transformers inherently outperform

LSTMs in sarcasm detection due to their superior contextual modeling and pretraining advantages. While attention mechanisms can further refine performance (XLNet with highest performance of 0.98), their utility depends on the base architecture. For practical applications, RoBERTa offers a balance of efficiency and accuracy, whereas XLNet + Attention represent the state-of-the-art for research benchmarks. Future work should address computational efficiency and multimodal integration to bridge the gap between laboratory performance and real-world usability.

8 Limitations and Future Work

This study's primary limitations include (i) the computational cost of fine-tuning large transformer models; (ii) the moderate dataset size, which restricts generalizability; and (iii) the absence of multimodal sarcasm analysis that incorporates audio or visual cues. Future work will address these constraints by leveraging multimodal fusion frameworks and exploring lightweight transformer variants for efficient deployment.

Acknowledgement: Authors are thankful to Dr. Neha Kushe for the technical support provided in carrying out this research work.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Study conception: Palak Bari and Anupama Jawale; data collection: Gurnur Bedi and Khushi Joshi; analysis and interpretation of results: Anupama Jawale and Palak Bari; drafting manuscript: Palak Bari, Gurnur Bedi, Khushi Joshi and Anupama Jawale. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data is available from authors on request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Riloff E, Qadir A, Surve P, De Silva L, Gilbert N, Huang R. Sarcasm as contrast between a positive sentiment and negative situation. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing; 2013 Oct 18–21; Seattle, WA, USA. p. 704–14. doi:10.18653/v1/d13-1066.
2. Brentari D, González C, Seidl A, Wilbur R. Sensitivity to visual prosodic cues in signers and nonsigners. Lang Speech. 2011;54(1):49–72. doi:10.1177/0023830910388011.
3. Poria S. Soujanyaporia/MUStARD [Internet]. 2025 [cited 2025 May 10]. Available from: https://github.com/soujanyaporia/MUStARD.
4. Chimote AK. An approach to detect sentence level sarcasm using deep learning techniques. Biosci Biotech Res Comm. 2020;13(14):125–8. doi:10.21786/bbrc/13.14/30.
5. Rahaman A, Kuri R, Islam S, Hossain MJ, Kabir MH. Sarcasm detection in tweets: a feature-based approach using supervised machine learning models. Int J Adv Comput Sci Appl. 2021;12(6):454–60. doi:10.14569/ijacsa.2021.0120651.
6. Chaudhari P, Chandankhede C. Literature survey of sarcasm detection. In: Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET); 2017 Mar 22–24; Chennai, India. p. 2041–6. doi:10.1109/WiSPNET.2017.8300120.
7. Sarsam SM, Al-Samarraie H, Alzahrani AI, Wright B. Sarcasm detection using machine learning algorithms in Twitter: a systematic review. Int J Mark Res. 2020;62(5):578–98. doi:10.1177/1470785320921779.
8. Parameswaran P, Trotman A, Liesaputra V, Eyers D. Detecting the target of sarcasm is hard: really? Inf Process Manag. 2021;58(4):102599. doi:10.1016/j.ipm.2021.102599.

9. Kumar A, Garg G. Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets. J Ambient Intell Humaniz Comput. 2023;14(5):5327–42. doi:10.1007/s12652-019-01419-7.
10. Rahaman Wahab Sait A, Khairi Ishak M. Deep learning with natural language processing enabled sentimental analysis on sarcasm classification. Comput Syst Sci Eng. 2023;44(3):2553–67. doi:10.32604/csse.2023.029603.
11. Goel P, Jain R, Nayyar A, Singhal S, Srivastava M. Sarcasm detection using deep learning and ensemble learning. Multimed Tools Appl. 2022;81(30):43229–52. doi:10.1007/s11042-022-12930-z.
12. Potamias RA, Siolas G, Stafylopatis AG. A transformer-based approach to irony and sarcasm detection. Neural Comput Appl. 2020;32(23):17309–20. doi:10.1007/s00521-020-05102-3.
13. Kumar S, Mondal I, Akhtar MS, Chakraborty T. Explaining (sarcastic) utterances to enhance affect understanding in multimodal dialogues. arXiv:2211.11049. 2022. doi:10.48550/arxiv.2211.11049.
14. Gupta S, Shah A, Shah M, Syiemlieh L, Maurya C. FiLMing multimodal sarcasm detection with attention. arXiv:2110.00416. 2021. doi:10.48550/arXiv.2110.0041.
15. Shangipour Ataei T, Javdan S, Minaei-Bidgoli B. Applying transformers and aspect-based sentiment analysis approaches on sarcasm detection. In: Proceedings of the Second Workshop on Figurative Language Processing; 2020 Jul 9; Online. p. 67–71. doi:10.18653/v1/2020.figlang-1.9.
16. Zhang M, Chang K, Wu Y. Multi-modal semantic understanding with contrastive cross-modal feature alignment. arXiv:2403.06355. 2024. doi:10.48550/arxiv.2403.06355.
17. Amin MM, Mao R, Cambria E, Schuller BW. A wide evaluation of ChatGPT on affective computing tasks. arXiv:2308.13911. 2023. doi:10.48550/arxiv.2308.13911.
18. Kumar S, Kulkarni A, Akhtar MS, Chakraborty T. When did you become so smart, oh wise one?! Sarcasm explanation in multi-modal multi-party dialogues. arXiv:2203.06419. 2022. doi:10.48550/arxiv.2203.06419.
19. Castro S, Hazarika D, Pérez-Rosas V, Zimmermann R, Mihalcea R, Poria S. Towards multimodal sarcasm detection (an_Obviously_perfect paper). arXiv:1906.01815. 2019. doi:10.48550/arxiv.1906.01815.
20. Kolchinski YA, Potts C. Representing social media users for sarcasm detection. arXiv:1808.08470. 2018. doi:10.48550/arxiv.1808.08470.
21. Amir S, Wallace BC, Lyu H, Silva PCMJ. Modelling context with user embeddings for sarcasm detection in social media. arXiv:1607.00976. 2016. doi:10.48550/arxiv.1607.00976.
22. Kumar Y, Goel N. AI-based learning techniques for sarcasm detection of social media tweets: state-of-the-art survey. SN Comput Sci. 2020;1(6):318. doi:10.1007/s42979-020-00336-3.
23. Marriwala NK, Shukla VK, William P, Guleria K, Sobti R, Sharma S. Detection of viral messages in twitter using context-based sentiment analysis framework. Int J Inf Technol. 2024;16(8):5069–75. doi:10.1007/s41870-024-02084-6.
24. Poria S, Cambria E, Hazarika D, Vij P. A deeper look into sarcastic tweets using deep convolutional neural networks. arXiv:1610.08815. 2016. doi:10.48550/arXiv.1610.08815.
25. Bansal S, Garimella V, Suhane A, Patro J, Mukherjee A. Code-switching patterns can be an effective route to improve performance of downstream NLP applications: a case study of humour, sarcasm and hate speech detection. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; 2020 Jul 5–10; Online. p. 1018–23. doi:10.18653/v1/2020.acl-main.96.
26. Poria S, Cambria E, Hazarika D, Majumder N, Zadeh A, Morency LP. Context-dependent sentiment analysis in user-generated videos. In: Proceedings of the 55th Annual Meeting of the Association forComputational Linguistics; 2017 Jul 30–Aug 4; Vancouver, BC, Canada. p. 873–83. doi:10.18653/v1/p17-1081.
27. Villegas DS, Preoţiuc-Pietro D, Aletras N. Improving multimodal classification of social media posts by leveraging image-text auxiliary tasks. arXiv:2309.07794. 2024. doi:10.48550/arXiv.2309.07794.
28. Grover K, Phaneendra Angara SM, Akhtar MS, Chakraborty T. Public wisdom matters! discourse-aware hyperbolic Fourier co-attention for social-text classification. arXiv:2209.13017. 2022. doi:10.48550/arxiv.2209.13017.