# Leveraging Machine Learning and BERT for Sarcasm Detection in Text

Kogilavani Shanmugavadivel[1], Priyadharshini C[2], Varshini L[3,*] and Sathyaa S[4]

*Department of AI, Kongu Engineering College, Perundurai, Erode*

## Abstract

Sarcasm detection in Tamil-English code-mixed text presents a unique challenge, particularly when traditional machine learning models are employed. This paper explores the application of conventional algorithms such as random forest, logistic regression, and naive Bayes, as well as the transformer-based BERT model. Performance evaluation uses four datasets, focusing on key metrics such as accuracy, precision, recall, and F1-score. BERT demonstrates superior performance, effectively capturing contextual nuances in sarcasm detection, making it a more viable approach for multilingual and code-mixed environments. Future work may expand on these findings by utilizing advanced transformer architectures and incorporating additional features like sentiment and emoji-based cues.

## Keywords

Sarcasm detection, Machine learning, BERT, Code-mixed text, Tamil-English, Transformer models, Natural Language Processing

## 1. INTRODUCTION

Sarcasm detection in the text is a challenging task, particularly in Tamil-English code-mixed social media text, where contextual and cultural nuances play a significant role. Sarcasm, often marked by a contrast between literal and intended meanings, requires models that can understand context, tone, and subtle linguistic cues. Conventional machine learning models such as Naive Bayes, Logistic Regression, and Random Forest have been widely used in text classification tasks but struggle to capture sarcasm due to their reliance on surface-level features like n-grams and bag-of-words. These methods often fail to model the deeper context necessary for sarcasm detection, especially in multilingual settings. Recent advancements in NLP, especially transformer-based models like BERT (Bidirectional Encoder Representations from Transformers), have shown remarkable success in understanding context and semantics, making them ideal for tasks like sarcasm detection. BERT's bi-directional attention mechanism allows it to better capture the nuanced expressions in Tamil-English code-mixed text. We aims to compare the performance of traditional machine learning models with BERT for sarcasm detection, using four datasets of code-mixed text. The performance of the models is evaluated using metrics such as accuracy, precision, recall, and F1-score.

## 2. LITERATURE SURVEY

Recent research has extensively leveraged machine learning and deep learning models to enhance sarcasm detection in text.

Chakravarthi et al. [1] present a comprehensive review of sarcasm detection in Dravidian languages using the DravidianCodeMix framework, focusing on the complexities of handling code-mixed social media text for linguistic analysis.

Kumar et al. [2] utilized Long Short-Term Memory (LSTM) models with CNN layers to capture intricate sarcasm patterns, demonstrating improved results in sarcastic comment classification.

*Forum for Information Retrieval Evaluation, December 12-15, 2024, India*

[*] Corresponding author.

kogilavani.sv@gmail.com (K. Shanmugavadivel); priyadharshinic.22aid@kongu.edu (P. C); varshinil.22aid@kongu.edu (V. L); sathyaas.22aid@kongu.edu (S. S)

BERT-based models have gained significant traction, as demonstrated by Khatri et al. [3] who applied BERT and GloVe embeddings to Twitter data, highlighting the superior performance of pre-trained transformers in sarcasm detection.

Multitask learning has also been explored, with Majumder et al. [4] focusing on combining sentiment analysis and sarcasm detection, showing how shared learning improves performance in both areas.

Similarly, Hiai and Shimada [5] employed Recurrent Neural Networks (RNNs) with relation vectors, enabling more nuanced detection of sarcastic patterns.

Hybrid models, such as those developed by Jain et al. [6] combined CNN and LSTM for enhanced sarcasm detection, effectively handling complex text structures.

Kumar and Garg [7] compared traditional machine learning models with advanced deep learning architectures, finding that deep models like BERT outperformed others in accuracy and generalization.

Multimodal approaches have further advanced the field. Poria et al. [8] integrated textual, visual, and audio features to detect sarcasm, achieving better results in multimedia contexts.

Ghosh et al. [9] tackled sarcasm detection in code-mixed languages, addressing the challenges of detecting sarcasm in multilingual conversations and showing how BERT can be effective in such contexts.

Zhang et al. [10] implemented BERT for sarcasm detection in tweets, emphasizing the role of context and the effectiveness of transformer models in processing social media language.

Eke et al. [11] introduced a context-based feature extraction technique for sarcasm identification using deep learning models, specifically leveraging BERT for improved accuracy on benchmark datasets. Their approach highlighted the importance of contextual understanding in sarcasm detection tasks.

Pandey and Singh [12] developed a BERT-LSTM model to detect sarcasm in code-mixed social media posts. The study addressed challenges posed by mixed languages, demonstrating that BERT's contextual embeddings combined with LSTM effectively capture sarcasm nuances.

Sandor and Babac [13] used machine learning techniques to detect sarcasm in online comments. Their work focused on building effective classifiers for sarcasm detection, leveraging multiple feature extraction methods to handle the complexity of sarcastic language online.

Kumar and Sarin [14] introduced WELMSD, a sarcasm detection approach combining word embeddings and language models. Their study demonstrated the effectiveness of this hybrid model in handling sarcasm through contextual understanding and word representations.

Goel et al. [15] explored sarcasm detection using deep learning and ensemble learning techniques. Their approach combined multiple models to improve classification performance, showcasing the potential of ensemble methods in enhancing sarcasm detection accuracy in multimedia applications.

Jeremy et al. [16] investigated sarcasm detection by optimizing various input methods in text data. Their work highlights the impact of input representations on model performance, emphasizing the importance of preprocessing and feature engineering in improving sarcasm classification accuracy.

Chakravarthi et al. [17] addressed sarcasm detection in Dravidian languages using the Dravidian-CodeMix dataset. They explored machine learning and deep learning methods, focusing on overcoming challenges in code-mixed and low-resource text.

## 3. METHODOLOGY

This section describes the approach taken to identify sarcasm in code-mixed Tamil-English text by utilising the transformer-based BERT model in addition to conventional machine learning models. Data preprocessing, model implementation, and evaluation are some of the crucial steps in the process.

### 3.1. DATA COLLECTION AND DATA PREPROCESSING

The dataset for sarcasm detection is composed of multiple files, each serving a specific purpose in the analysis process. The first dataset, 'Change Makers Tamil.csv', contains 47,960 rows and two columns: 'Id' and 'Labels'. The 'Labels' column identifies whether the content is sarcastic or non-sarcastic. This dataset provides labels necessary for training models on identifying sarcasm.

The second dataset, ‘sarcasm tam dev.csv‘, includes text data in a multilingual format (Tamil and English), with two columns: ‘Text‘ and ‘Labels‘. The ‘Text‘ column contains the actual social media content collected from platforms like Facebook and Twitter, and the ‘Labels‘ column classifies each text as either sarcastic or non-sarcastic.

The third dataset, ‘sarcasm tam test without labels.csv‘, contains only the ‘ID‘ and ‘Text‘ columns, representing the test set where labels are not provided. This dataset will be used for evaluating the model’s ability to predict sarcasm.

The fourth dataset, ’sarcasm tam train.csv’, comprises both ’Text’ and ’Labels’ columns , serving as the training data for the model. This dataset will be utilized to train the model to differentiate between sarcastic and non-sarcastic text, ensuring it learns the patterns for sarcasm detection in the Tamil language.

In preprocessing, several steps were applied to clean and prepare the text for model training. Tokenization was performed while retaining both Tamil and English tokens. Text was converted to lowercase, and transliterated Tamil words were normalized. Special characters, digits, and punctuation were removed to reduce noise in the data. Additionally, stopwords from both Tamil and English were filtered out to focus on meaningful content.

Sequence padding was applied to ensure uniform input lengths for the models. Contextual embeddings were generated using BERT to capture nuanced meaning from the text, while more traditional models utilized GloVe and Word2Vec embeddings. Finally, the dataset was split into training and testing sets (80-20 split), with cross-validation applied to ensure robust model evaluation. This preprocessing pipeline ensured the data was clean, consistent, and ready for sarcasm detection models.

## 3.2. Bidirectional Encoder Representation from Transformers

BERT (Bidirectional Encoder Representations from Transformers) can interpret context and pick up on little linguistic clues, it is essential for sarcasm detection. In contrast to conventional models, BERT makes use of a transformer-based architecture with a bidirectional attention mechanism, which allows it to take into account the words that come before and after one another in a phrase at the same time. This bi-directional feature is essential for sarcasm recognition, particularly in code-mixed Tamil-English language where sarcastic statements must be understood in context.

Since sarcasm frequently relies on opposing literal and intended interpretations, BERT’s pre-trained embeddings capture deep contextual links between words, which makes it particularly successful in identifying sarcasm. By honing BERT’s performance on the sarcasm detection task, one can help it better distinguish between sarcastic and non-sarcastic comments by teaching it task-specific subtleties in code-mixed text. In comparison to standard models, the BERT-based model outperforms them in capturing the complexity of multilingual sarcasm, as evidenced by its better accuracy and F1-score. Performance measurements show that BERT performs significantly better than models such as Random Forest and Naive Bayes. The BERT classification report is shown in Table 1.

Beyond its efficiency, BERT’s adaptability makes it scalable and suitable for real-world applications since it can be optimised for a variety of natural language processing jobs. BERT is a recommended option for text classification tasks like sarcasm detection because of its strong design and capacity to handle large-scale datasets.

**Table 1**
Classification Report for BERT

| **Metrics** | ***Value*** |
|---|---|
| Accuracy | 80.23 |
| Precision | 74.94 |
| Recall | 73.63 |
| F1-score | 74.22 |

Formula : In the BERT architecture, parameters are the quantity of attention heads, hidden units, and layers. BERT is incredibly successful in sarcasm recognition because the embedding layer uses token, positional, and segment embeddings to capture word associations and attention layers model contextual dependencies.

## 3.3. Random Forest

By aggregating the predictions of several decision trees, Random Forest is an ensemble learning technique that is important for sarcasm identification. To increase generalisation and decrease overfitting, each tree in the forest is trained using a different random subset of the features and data. As they combine the results from different trees to produce more reliable and precise predictions, Random Forests do exceptionally well in complicated classification tasks.

Random Forest uses its capacity to identify patterns in the data to detect sarcasm by taking into account several word attributes including frequency and word pairings (n-grams). Despite its reliance on surface-level features such as TF-IDF vectors and bag-of-words, Random Forest is a strong foundation for sarcasm detection, especially in difficult code-mixed Tamil-English language, since it can handle the non-linear correlations between these features.

The Random Forest model struggles to capture the deeper contextual details that are essential for sarcasm detection, which makes it less accurate than deep learning techniques like BERT. Nevertheless, it still achieves a respectable level of accuracy. However, it is a useful tool in many text classification jobs due to its ease of use, interpretability, and capacity to handle big datasets with little adjustment.The Random Forest model's classification report is displayed in Table 2.

**Table 2**
Classification Report for Random Forest

| **Metrics** | ***Value*** |
|---|---|
| Accuracy | 78 |
| Precision | 74 |
| Recall | 66 |
| F1-score | 68 |

Formula : The number of trees in the forest and the depth of each tree determine how many parameters there are in Random Forest. The model can achieve better generalisation than a single decision tree since each decision tree is constructed using a portion of the data and characteristics, and the final result is the majority vote or average of all tree forecasts.

## 3.4. Logistic Regression

Logistic Regression is a popular machine learning model,because of its interpretability and effectiveness in binary classification problems, for sarcasm detection. It is straightforward yet effective. It works by estimating the likelihood that an input, given its attributes and the goal label, would fall into a particular class—in this example, sarcastic or non-sarcastic. The output probabilities of logistic regression are subjected to a sigmoid function before being thresholded to provide binary predictions.

Word frequencies, n-grams, and TF-IDF vectors are some of the variables that Logistic Regression utilises to categorise text in order to detect sarcasm in Tamil-English code-mixed text. The deeper contextual and non-linear linkages that are frequently essential for sarcasm identification can be difficult for Logistic Regression to capture, even though it is excellent at handling linearly separable data. This is especially true in multilingual literature where meaning is dependent on nuanced linguistic cues.

Because of its simplicity and efficiency, Logistic Regression performs fairly well as a baseline model despite its limits in addressing complicated patterns; it achieves moderate accuracy in many classification tasks. It is especially helpful for short assessments and in situations where processing power is scarce.The classification report for logistic regression is displayed in Table 3.

**Table 3**
Classification Report for Logistic Regression

| **Metrics** | ***Value*** |
|---|---|
| Accuracy | 79 |
| Precision | 75 |
| Recall | 66 |
| F1-score | 68 |

Formula : The weights assigned to each feature in logistic regression are the parameters, and they are discovered by gradient descent in order to minimise the log loss function. By reflecting each feature's contribution to the classification, these weights enable the model to forecast based on the linear relationship between the target output and the input data.

### 3.5. Naive Bayes

Because of its effectiveness and simplicity, the probabilistic machine learning model Naive Bayes is frequently employed for sarcasm detection. The model is computationally efficient because it applies Bayes' Theorem under the assumption that all characteristics (words) are independent, which is rarely the case in real-world data. By multiplying the conditional probabilities of each word in the text with a class label, Naive Bayes determines if a text is sardonic or not. For text classification tasks such as sarcasm detection, it frequently achieves good results, despite the naive independence assumption.

Word frequencies and TF-IDF scores are two examples of features that Naive Bayes utilises to classify text in sarcasm detection for Tamil-English code-mixed text. It performs best when sarcasm is expressed using particular terms or patterns that can be statistically distinguished from non-sarcastic classes. However, compared to more sophisticated models like BERT, its independence assumption restricts its capacity to capture intricate contextual links, making it less successful for subtle sarcasm detection.

Naive Bayes is a good choice for first phase classification jobs since it is simple to use, computationally quick, and performs quite well on limited datasets or as a baseline model.The Naive Bayes classification report is displayed in Table 4.

**Table 4**
Classification Report for Naive Bayes

| **Metrics** | ***Value*** |
|---|---|
| Accuracy | 79 |
| Precision | 74 |
| Recall | 69 |
| F1-score | 71 |

Formula:The overall class probabilities and the conditional probabilities of each feature (word) given a class label are the parameters used in Naive Bayes. By multiplying the likelihood of each class's features by the class's prior probability, the model predicts the class with the highest posterior probability.

## 4. RESULTS AND DISCUSSIONS

The results reveal that the BERT model significantly outperforms traditional machine learning models Random Forest, Logistic Regression, and Naive Bayes in detecting sarcasm in Tamil English code mixed text. BERT achieved higher accuracy, precision, recall, and F1-score across four datasets, showcasing its ability to capture the nuanced context and cultural subtleties of sarcasm. In contrast, traditional models, which rely on surface level features, struggled to recognize these complexities. The success of BERT can be attributed to its transformer architecture, which employs bi-directional attention to analyze word relationships in context. This indicates that advanced models like BERT are more effective for sarcasm detection in multilingual environments, and future efforts should consider incorporating features like sentiment analysis and emoji recognition for further improvements.

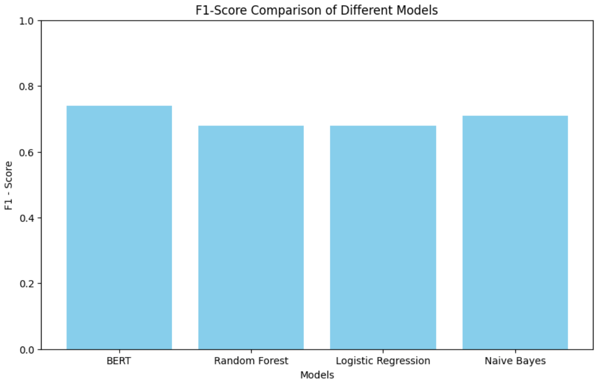

**Figure 1:** Accuracy

Figure 1 shows the accuracy comparison of the algorithms used in this study, including BERT, Random Forest, Logistic Regression and Naive Bayes models. It highlights the superior performance of BERT, which achieved the highest accuracy in sarcasm detection in text. This comparison enables a clear assessment of each model's effectiveness.

## 5. CONCLUSION

In conclusion, the findings highlight the effectiveness of BERT for sarcasm detection in Tamil-English code-mixed text, demonstrating its superiority over traditional machine learning algorithms. The results emphasize the need for advanced models that can understand contextual and cultural nuances. Future research should explore additional transformer architectures and incorporate features such as sentiment analysis to enhance detection accuracy even further. The insights gained can significantly improve sarcasm detection systems in multilingual contexts.

## 6. FUTURE RESEARCH DIRECTIONS

Future research on sarcasm detection should explore advanced transformer architectures beyond BERT, such as RoBERTa and GPT, to enhance context understanding. Incorporating multimodal data, including audio and visual cues, can improve detection accuracy by capturing tone and facial expressions. Additionally, expanding datasets to include diverse languages and dialects will enhance model generalization. Integrating sentiment analysis and emoji recognition can refine sarcasm detection further. Finally, developing real-time systems for social media monitoring and conversational agents could translate research findings into practical applications, improving understanding of human communication.

## Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT in order to: drafting content, grammar and spelling check, etc. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

## References

[1] B. R. Chakravarthi, N. Sripriya, B. Bharathi, K. Nandhini, S. C. Navaneethakrishnan, T. Durairaj, R. Ponnusamy, P. K. Kumaresan, K. K. Ponnusamy, C. Rajkumar, Overview of sarcasm identification of dravidian languages in dravidiancodemix@ fire-2023, in: FIRE (Working Notes), 2023.

[2] A. Kumar, S. R. Sangwan, A. Arora, A. Nayyar, M. Abdel-Basset, et al., Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network, IEEE access 7 (2019) 23319–23328.

[3] A. Khatri, et al., Sarcasm detection in tweets with bert and glove embeddings, arXiv preprint arXiv:2006.11512 (2020).

[4] N. Majumder, S. Poria, H. Peng, N. Chhaya, E. Cambria, A. Gelbukh, Sentiment and sarcasm classification with multitask learning, IEEE Intelligent Systems 34 (2019) 38–43.

[5] S. Hiai, K. Shimada, Sarcasm detection using rnn with relation vector, International Journal of Data Warehousing and Mining (IJDWM) 15 (2019) 66–78.

[6] D. Jain, A. Kumar, G. Garg, Sarcasm detection in mash-up language using soft-attention based bi-directional lstm and feature-rich cnn, Applied Soft Computing 91 (2020) 106198.

[7] A. Kumar, G. Garg, Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets, Journal of ambient intelligence and humanized computing 14 (2023) 5327–5342.

[8] S. Poria, D. Hazarika, N. Majumder, R. Mihalcea, Beneath the tip of the iceberg: Current challenges and new directions in sentiment analysis research, IEEE transactions on affective computing 14 (2020) 108–132.

[9] S. Ghosh, S. Ghosh, D. Das, Sentiment identification in code-mixed social media text, arXiv preprint arXiv:1707.01184 (2017).

[10] Y. Zhang, D. Ma, P. Tiwari, C. Zhang, M. Masud, M. Shorfuzzaman, D. Song, Stance-level sarcasm detection with bert and stance-centered graph attention networks, ACM Transactions on Internet Technology 23 (2023) 1–21.

[11] C. I. Eke, A. A. Norman, L. Shuib, Context-based feature technique for sarcasm identification in benchmark datasets using deep learning and bert model, IEEE Access 9 (2021) 48501–48518.

[12] R. Pandey, J. P. Singh, Bert-lstm model for sarcasm detection in code-mixed social media post, Journal of Intelligent Information Systems 60 (2023) 235–254.

[13] D. Šandor, M. B. Babac, Sarcasm detection in online comments using machine learning, Information Discovery and Delivery (2023).

[14] P. Kumar, G. Sarin, Welmsd–word embedding and language model based sarcasm detection, Online Information Review 46 (2022) 1242–1256.

[15] P. Goel, R. Jain, A. Nayyar, S. Singhal, M. Srivastava, Sarcasm detection using deep learning and ensemble learning, Multimedia Tools and Applications 81 (2022) 43229–43252.

[16] N. Jeremy, M. Alam, J. T. Tirtawijaya, J. S. Lumentut, Optimizing sarcasm detection through various input methods in text data, in: 2024 2nd International Conference on Technology Innovation and Its Applications (ICTIIA), IEEE, 2024, pp. 1–5.

[17] B. R. Chakravarthi, S. N, B. B, N. K, T. Durairaj, R. Ponnusamy, P. K. Kumaresan, K. K. Ponnusamy, C. Rajkumar, Overview of sarcasm identification of dravidian languages in dravidiancodemix@fire-2024, in: Forum of Information Retrieval and Evaluation FIRE - 2024, DAIICT , Gandhinagar, 2024.