Sarcasm Unveiled: Advanced Detection Techniques for Tamil and Malayalam Using Multi modal Approaches*

Sumathi S[1,*,†], Jayaseelan S[2,†] and Kevin Jeyaraj P[3,†]

[1] *Department of Information Technology, St.Joseph's College of Engineering*
[2] *Department of Artificial Intelligence and Machine Learning,St.Joseph's College of Engineering*
[3] *Department of Artificial Intelligence and Machine Learning,St.Joseph's College of Engineering*

Abstract

In low-resource languages like Tamil and Malayalam, sarcasm detection poses numerous challenges due to linguistic complexities and the intricate blending of languages. This paper proposes a novel approach for sarcasm detection in Tamil and Malayalam texts using a deep learning-based BERT model. A specially curated corpus, containing sentences tagged with sarcasm in these languages, is employed to train the model. We ensured that all irrelevant characters, symbols, and links were excluded, allowing the model to process only the clean, language-specific data. The multilingual BERT tokenizer was used to tokenize the dataset, followed by BERT for Sequence Classification to train the model. Sarcastic and non-sarcastic statements were successfully classified. A weighted cross-entropy loss function along with the AdamW optimizer and a linear learning rate scheduler was utilized during training. Results demonstrated significant improvements in accuracy and macro F1-score across multiple validation sets, with a maximum F1 score of 0.736 for Tamil and 0.725 for Malayalam. While this method holds significant potential for deployment in various domains certain applications such as social analysis and user dissatisfaction with government services may raise concerns about censorship in user activities. Nevertheless, this model offers an invaluable tool for governments and organizations to more accurately analyse public sentiment and engagement, aiding in informed decision-making and policy improvements.

Keywords

Sarcasm detection, BERT, Natural Language Processing, Multilingual Transformer Based Model, Deep Learning, Sentiment Analysis

1. Introduction

The proliferation of social media platforms and online discussion forums has significantly increased the complexity of natural language processing (NLP) tasks, especially in the domain of sentiment analysis and sarcasm detection. Sarcasm, characterized by its subtle linguistic nuances, often poses challenges to NLP models, particularly in low-resource languages like Tamil, Tanglish (a blend of Tamil and English), and Malayalam. These languages frequently involve code-mixing and unique idiomatic expressions, making it difficult for conventional models to capture the underlying sentiment accurately. This difficulty is further compounded by the limited availability of annotated data for these languages, hampering the effectiveness of traditional machine learning models. The task of sarcasm detection becomes crucial in applications such as sentiment analysis for government services, monitoring customer feedback, and moderating user content on social media platforms. Incorrectly interpreting sarcasm can lead to misguided decisions, as sarcastic comments might be misclassified as positive or negative sentiments. This paper aims to address these challenges by introducing a novel approach to sarcasm detection using deep learning-based models, specifically leveraging the multilingual BERT framework. BERT (Bidirectional Encoder Representations from Transformers) provides a robust structure for understanding the context of sentences, even in complex and mixed-language settings. The proposed approach utilizes the capabilities of BERT to process a corpus of Tamil, Tanglish, and Malayalam texts, enabling the model to recognize and classify sarcastic versus non-sarcastic statements

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

[*] Corresponding author.

[†] These authors contributed equally.

sumathiselvan79@gmail.com (S. S); jayaseelansrinivasan00@gmail.com (J. S); kevinjeyaraj04@gmail.com (K.J. P)
0000-0001-5635-2468 (S. S)

effectively. By employing techniques such as data cleansing, tokenization, and fine-tuning with a sequence classification head, we ensure that the model captures the intricate patterns of sarcasm. This approach not only enhances the accuracy of sarcasm detection but also enables a more reliable analysis of user sentiments, providing valuable insights for organizations and governments in shaping communication strategies and improving user engagement.

2. Literature survey

Extensive study was conducted which focused on detecting abusive language and analyzing sentiment in Tamil and Malayalam, two under-resourced languages[1] . The research addressed the growing need for multimodal detection in social media, employing multiple deep-learning models, including mBERT for text, ViT for images, and MFCC for audio. The approach secured the top rank by achieving a weighted F1 score of 0.5786 for abusive language detection and notable results in sentiment analysis for both Tamil and Malayalam. Comprehensive review on sarcasm detection was done which emphasize its importance in sentiment analysis. [2] Challenges of identifying sarcasm was highlighted which often eludes traditional sentiment detection models. The study explored various approaches, including linguistic resources and machine-learning classifiers, focusing on detecting sarcasm in Telugu conversational data. Deep learning learning was utilized which bridged the gap between sentiment and sarcastic expression, particularly in the context of political and hyperbolic statements. The review also emphasized pre-processing techniques to enhance model accuracy.

An efficient approach was proposed for detecting sarcasm in tweets by analyzing polarity flips.[3] The study highlights the complexities of identifying sarcasm, particularly on platforms like Twitter, where users often employ figurative language. By using 21 features focusing on context, contrast, and emotions, the model demonstrated that sarcastic tweets tend to have more polarity shifts than non-sarcastic ones. Machine learning classifiers, such as MLP and Random Forest, showed superior performance, achieving a high accuracy rate of 94%, offering valuable insights for text mining and sentiment analysis. A novel approach combining Chameleon Swarm Optimization (CSO) with machine learning for sarcasm detection and sentiment classification was introduced. [4] The CSOML-SASC model incorporates pre-processing, TF-IDF-based feature extraction, and a weighted kernel extreme learning machine (WKELM) for classification, enhanced through parameter optimization. The study addressed the complexities of sarcasm, where positive vocabulary often conveys negative sentiments. The technique demonstrated superior performance over other recent models in detecting sarcasm on benchmark datasets, offering a robust solution to improve sentiment analysis in social media contexts. [5] presented a cross-linguistic sarcasm detection approach which targeted code-mixed Tamil and Malayalam comments from YouTube. The research was developed for the Dravidian-CodeMix FIRE 2023 task, utilized various machine learning algorithms, such as Count Vectorizer and TF-IDF with classifiers like MLP and Random Forest. The Tamil model ranked second, while the Malayalam model topped the competition. With validation accuracies ranging from 0.72 to 0.85, this study highlights the efficacy of these methods in addressing challenges posed by code-mixing and linguistic diversity in online content.

Hate speech and offensive language detection in CodeMix Dravidian languages using a cost-sensitive learning approach.[6] have been explored. Multilingual transformer-based models like BERT, MuRIL, and LaBSE across Kannada-English, Malayalam-English, and Tamil-English datasets have been evaluated which achieved 96% accuracy for Malayalam and 72% for Tamil. The study also highlighted MuRIL with SVM and RBF kernel as a consistent performer. A cost-sensitive learning strategy was applied to tackle class imbalance, significantly improving model accuracy and reducing bias across all datasets. [7]focused on meme classification for the Tamil language which considering only the text embedded in the meme, rather than analyzing the image itself. The authors proposed a machine learning approach to address the challenge of trolling, offensive content, and sarcasm often hidden within memes. The multi-modal complexity of memes, especially in Tamil, is highlighted, and their approach achieves an F1 score of 0.50 on the test set during the DravidianLangTech-EACL2021 task.

Findings of a shared task [8] are outlined which focused on multimodal abusive language detection and sentiment analysis in Tamil and Malayalam conducted during the RANLP 2023 workshop. The authors emphasized the importance of this task for developing models capable of analyzing abusive content and fine-grained sentiment from diverse multimodal data—comprising video, audio, and text.

A study on sarcasm detection in Dravidian languages using the ALBERT transformer model.[9] The study focused on classifying text as sarcastic or non-sarcastic in Tamil and Malayalam languages. Through training and testing, the model achieved macro F1 scores of 0.48 for Tamil and 0.34 for Malayalam, highlighted the challenges in detecting sarcasm within these languages. [10] explored sentiment analysis in Malayalam using machine learning classifiers which addressed the challenges associated with the language's low resource status. The study presented models like SVM, Naïve Bayes, and Random Forest, achieving notable results in classifying sentiments in social media data.

Conducted a comprehensive review of sarcasm detection techniques over multilingual platforms [11] was conducted with a focus on Dravidian languages like Tamil and Malayalam. The study highlighted the complexities of using machine learning and deep learning strategies, especially Recurrent Neural Networks (RNNs), for detecting sarcasm in code-mixed texts. It also analyzed the effectiveness of different models on datasets derived from social media, emphasizing the need for more nuanced models that can better understand the subtle nature of sarcasm. Sarcasm detection in code-mixed texts using transformer-based models [12] was explored which emphasize the importance of context in understanding sarcasm. The study utilized mBERT and MURIL models which addressed the challenges posed by code-mixed Tamil-English and Malayalam-English texts. The findings revealed that the MURIL model achieved impressive results, ranking first in the Tamil-English subtask with a Macro-F1 score of 0.781 and second in the Malayalam-English subtask with a score of 0.731, demonstrating the efficacy of transformer models in understanding nuanced language in social media contexts.

[13] investigated sarcasm detection in Tamil and Malayalam using various transformer architectures, such as IndicBERT, mBERT, and DistilBERT. The study highlighted the challenges posed by the unique linguistic structures and cultural nuances of these languages. The results indicated that IndicBERT achieved a Macro-F1 score of 0.82 for Tamil, while mBERT performed relatively better for Malayalam with a score of 0.66. This research emphasized the potential of transformer models to effectively capture sarcastic expressions in low-resource languages. BERT-based model for sarcasm detection in Tamil and Malayalam code-mixed texts,[14] was proposed which addressed the challenge of identifying verbal irony in social media comments. The model included additional neural network layers to improve classification accuracy. The experiments yielded an F1 score of 0.72 for both languages, showcasing the model's ability to handle the intricacies of sarcasm in these low-resource languages.

[15] explored sarcasm detection in Tamil-English and Malayalam-English code-mixed social media content. The study used a combination of TF-IDF Vectorizer and a stacking classifier, which achieved weighted average F1 scores of 0.79 for Tamil and 0.78 for Malayalam. The research highlighted the complexities of detecting sarcasm in multilingual contexts and demonstrated significant advancements in accurate classification for sentiment analysis tasks. Recursive Neural Network [16] was proposed to enhance sentiment analysis for Tamil content. The study aimed to improve the understanding of longer or more complex phrases, including those with sarcasm. By employing Recursive Neural Networks, the research focused on inter-sentential sentiment prediction, thus offering a more nuanced interpretation of Tamil sentiments compared to traditional approaches like Naïve Bayes.

[17]proposed a BERT-LSTM model for sarcasm detection in code-mixed social media posts, highlighting the complexities of detecting sarcasm due to the absence of clear indicators in code-mixed text. The model used a pre-trained BERT for generating embeddings, followed by an LSTM layer to classify sarcasm. This approach outperformed traditional models, achieving a significant F1-score improvement of up to 6%, showcasing its potential for this challenging domain. The detection of offensive language in code-mixed Tamil-English social media comments [18] was addressed. The study detailed their participation in the HASOC-Dravidian-CodeMix-FIRE 2021 competition, employing techniques like Multilingual BERT for feature extraction. The team achieved a macro F1-score of 0.84, demonstrating the effectiveness of their approach in managing the challenges of offensive language detection in code-mixed contexts. [19]explored the role of emojis in emotion detection for Tamil text, focusing on

the challenges of understanding sentiment in low-resource languages. The study utilized techniques such as TF-IDF and MuRIL embeddings to analyze the influence of emojis on sentiment detection. Their results revealed that incorporating emoji information improved emotion recognition, highlighting the importance of visual cues in sentiment analysis for Tamil text.

Sentiment in Malayalam social media posts [20] was analyzed, addressing the challenges of working with low-resource languages. The study employed models like SVM, Random Forest, and Naïve Bayes for sentiment classification. The work emphasized the need for more annotated datasets and improved pre-processing methods to increase classification accuracy in Malayalam. Sentiment and emotion detection in Malayalam YouTube comments using deep learning models like LSTM and GRU [21] was studied. The study provided insights into the complexities of emotion detection in Malayalam due to limited annotated data and the informal nature of YouTube comments. The proposed models demonstrated superior performance over traditional methods, achieving a notable improvement in accuracy.

3. Proposed Solution

The enhancement of sarcasm detection in Tamil and Malayalam languages is the objective of the proposed solution using improved machine learning techniques with more emphasis on the usage of BERT models for precise and better calculation. The Solution explains and addresses all aspects that include data preprocessing, model training, evaluation, and prediction to guarantee that the model performs optimally and consistently.

3.1. Key Features of the Proposed Solution

3.1.1. Data Preprocessing

Text Cleaning: The preprocessing phase is essential because it ensures that the information input into the model is clean and useful. In this case the preprocess_text function strips away any URL, special character, number or other character that is not TamilTanglish or combination of Malayalam and english so as to concentrate on relevant information more effectively. This step also eliminates irrelevant information in the data which is also a plus in enhancing the models performance.

Label Encoding: To make it possible to train the models with the labels, a LabelEncoder is used to convert the labels into numerical format. This step is crucial as it helps in transforming the categorical data into the form that can be used by the machine learning models.

3.1.2. Custom Dataset Class

The dataset for sarcasm detection is derived from the shared task outlined in the overview of DravidianCodeMix@FIRE-2024 (Chakravarthi et al., 2024). To facilitate the loading and processing of text data, a custom Dataset class for PyTorch has been implemented. This class divides the text into tokens, performs tokenization, and prepares the data to be sent to the BERT model. It also makes sure that the data comes with labels Attention masks and Input IDs, which are essential for training and testing the model.

3.1.3. BERT-Based Model

Model Architecture: SarcasmDetectionModel employs attention-based transformer model BERT, which is previously trained on multilingual corpus, as a backbone. To tailor the model a linear classification layer and a dropout layer is inserted to the network. BERT's word embedding models are effective in capturing the textual subtleties that are necessary in sarcasm.

Forward Pass: The forward pass of the model takes in the input text and generates predictions that are subsequently used for computing the losses and tense decisions in sarcasm detection.

3.1.4. Training and Evaluation

Training Function: The process of model training for one epoch is organized in a single function train_epoch that includes data retrieval and processing, passing the data through the network and computing the loss, as well as its derivatives to reach the desired state of the model. The training employs the AdamW optimizer and a learning rate scheduler that effectively vary the learning rate throughout the training process, facilitating improved convergence.

Evaluation Function: The eval_model function assesses how well the model performs on validation data by determining the accuracy and the F1 score. This function assists in tracking the performance of the model and provides a basis for making adjustments for better accuracy.

3.1.5. Model Performance

Training Metrics: The model underwent training for four epochs where performance metric losses and accuracies are recorded for each epoch. The training results illustrate the precision keeps increasing as the loss reduces quality meaning that, the model is learning as expected.

Validation Results: The ability of the model to detect sarcasm in new data is corroborated by the validation accuracy and F1 score. The validation accuracy is 0.7934, and the F1 score is 0.7257, which indicates that the model performs great, although some aspects need further improvement.

3.1.6. Testing and Prediction

Prediction Generation: After the process word embedding, a model will then be applied to the test data set in order to determine the level of sarcasm present. The test data is passed through the model to generate outcomes, and the outcomes are stored for future processes.

Submission: The conclusive projections are structured in a CSV layout so that they may be submitted online as required. Also ensuring that the said results are in the appropriate format for the assessment. The suggested approach utilizes the state of the art recently BERT based models for sarcasm detection in Tamil and Malayalam which is the solution's strength in dealing with many complex problems. Emphasis is placed on detailed data preprocessing, efficient model architecture, and comprehensive evaluation of the model with the aim of improving the accuracy and reliability in detecting sarcasm. The results are of reasonable encouraging performance level which makes this method a worthy addition to the area of sentiment analysis and sarcasm detection.

4. System Architecture

The sarcasm detection system architecture in both Tamil and Malayalam languages is built around a robust and well-structured model utilizing the cutting-edge pre-trained models. The two main ingredients of this architecture are the BERT model and the application of the BERT neural network to perform sarcasm classification. A brief description of the model architecture and pre-trained models is given below.

4.1. Model Architecture Overview

The architecture shown in Fig. 1 can be divided into several main parts. The `SarcasmDetectionModel` is the most important element of the set. This model uses the BERT (Bidirectional Encoder Representations from Transformers) architecture for classifying text into sarcastic and nonsarcastic comments.

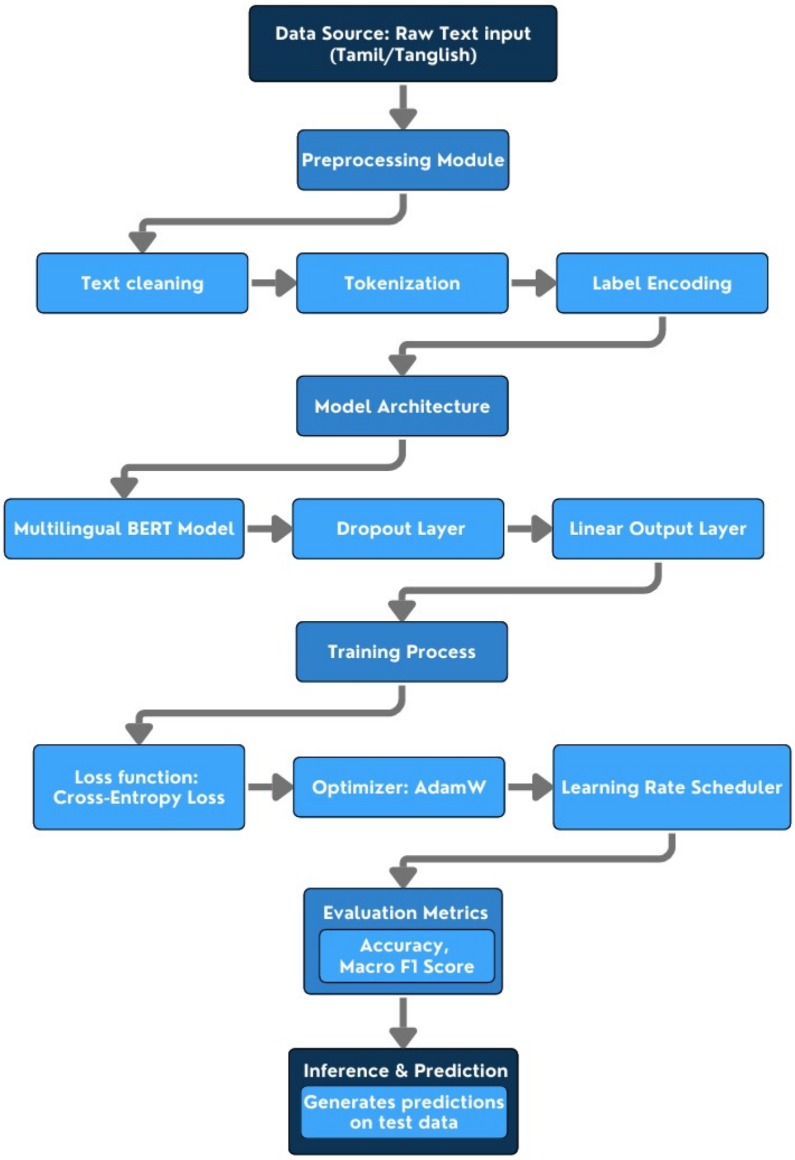

Figure 1: System Architecture

4.1.1. BERT Model

Pre-trained Transformer Model: The SarcasmDetectionModel employs BERT as its main model architecture and, in particular, uses bert base-multilingual cased. BERT is a large transformer-based model for natural language processing that is pre-trained on a corpus by understanding the meaning of a word in different contexts, that is, from both the left context of the target word and the right context.

Encoder Layers: BERT is made of multiple encoder layers (12 layers for bert-base) which are equipped with self-attention sublayers and feedforward neural networks. These layers enable the model to learn complex structures and relationships in the text.

Hidden States and Pooling: The contextualized embeddings are produced for each of the tokens in the input text. When performing classification tasks on the input sequence as a whole, the output of the [CLS] token from the last layer is used to summarize the input sequence.

4.1.2. Custom Layers

Dropout Layer: A dropout layer having a dropout rate of 0.3 has been incorporated into the model. By randomly eliminating a portion of the input units during training, dropout reduces the chances of overfitting. This regularization technique guarantees a good performance of the model even on new data.

Linear Output Layer: The conclusive linear layer projects the result of BERT's pooling layer into the number of target classes (which is sarcastic or non-sarcastic, in this case). This layer also reduces the high-dimensional output from BERT into a classification friendly format.

4.2. Training and Optimization

The model is trained using the following setup:

Optimizer: The model weights are updated using the AdamW optimizer. AdamW is a type of Adam optimizer which adds a weight decay mechanism to offset overfitting by imposing a penalty on high weight values. This optimizer is ideal for transformer-based models such as BERT.

Learning Rate Scheduler: The learning rate is controlled with the help of a linear schedule with warmup. This scheduler implements a learning rate policy, where one starts with a low learning rate, increases it to a maximum level, and reduces it afterwards according to the training stage. This method has been proven to help in stabilizing training and improving convergence.

Loss Function: The model's performance is gauged using cross-entropy loss. This loss function quantifies how much the predicted probabilities deviate from the true labels so that the model is corrected in its predictions.

4.3. Fine-Tuning BERT

BERT, which stands for Bidirectional Encoder Representations from Transformers, is the core of this research work. It was first trained on a vast amount of text to learn general language representations. Starting from scratch, the BERT model that has been pre-trained in Tamil and Tanglish and English and Malayalam is used to perform the task of detecting sarcasm in those languages. In this case, model fine-tuning refers to training the BERT model that has already been developed on a sarcasm detection task dataset with additional custom layers. This process also optimizes BERT's weights to enhance the model's ability to detect sarcasm in the specific target languages.

4.4. Evaluation and Prediction

The performance assessment of the model is conducted using two metrics: the accuracy and the macro F1-score. The accuracy refers to the ratio of the number of instances correctly classified to the total number of instances, and the F1score presents the performance of the model with respect to the precision and the recall. In the case of inference, the trained model predicts the output for the test data and stores them in a CSV file for submission. The Sarcasm Detection System in Tanglish and malalayam is based on the structure of the BERT model. BERT serves as a crucial component in understanding the complexities of textual content like sarcasm with its bidirectional attention mechanism and extensive pre–training. The architecture includes extra layers and optimization methods to adapt BERT to the task of sarcasm identification. Thus this pre-trained model is the one responsible for classifying sarcasm at a very high accuracy level meaning that the model is capable of managing the complexities involved with the Tamil ,Tanglish and malayalam languages.

5. Results and Discussions

5.1. Results for Tamil

The SarcasmDetectionModel was utilized based on BERT architecture with the core model being bert-base-multilingual-cased. The specifics of the modelling make up the training and evaluation procedures performed on the Tamil text dataset, and respective performance measures accuracy and F1-score are reported to show how well the model performs. The section describes the results of the model evaluation and discusses how well the model performed.

5.1.1. Model Performance

The training of the model was done for four epochs utilizing batches of 16 samples.

5.1.2. Graphical Representations

To provide a visual understanding of the model's performance, the following graphical representations are included

Training and Validation Accuracy Over Epochs: A visual representation in the form of a line graph displaying the evolution of accuracy metrics on the training and validation data sets during the course of the epochs.

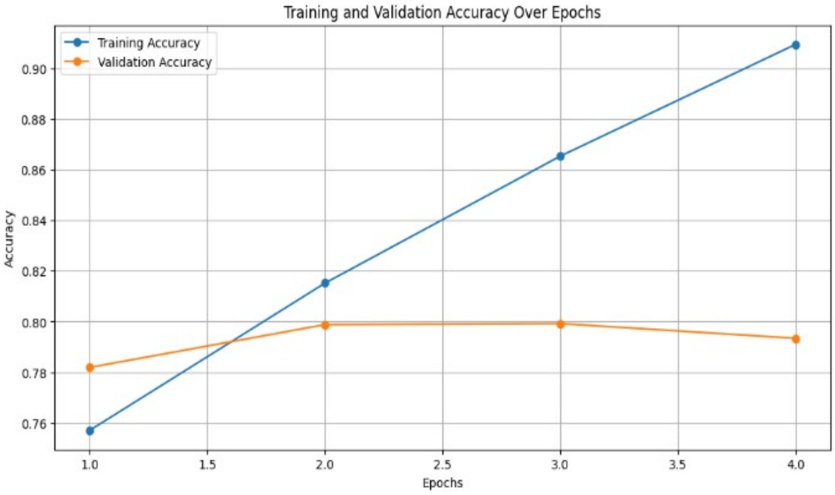

Figure 2: Training and Validation Accuracy Over Epochs

The graph in Fig. 2. displays Training and Validation Accuracy over Epochs as training progresses.

The training accuracy increases steadily with each epoch, starting from 75% and reaching approximately 90% by the fourth epoch. This consistent improvement indicates that the model is learning effectively from the training data.

The validation accuracy increases initially from about 70% but then plateaus and even decreases slightly after the second epoch. The initial increase indicates that the model's learning also benefits the validation data up to a point However, the plateau and subsequent decrease in accuracy in later epochs suggest that the model's performance on unseen data is not improving further, despite its increased accuracy on training data.

Training and Validation Loss Over Epochs: A line graph depicting the loss values across epochs with respect to both training and validation data sets.

The graph in Fig. 3. shows that the training loss decreases consistently over the epochs, indicating better learning of the training data, while the validation loss initially decreases but then increases

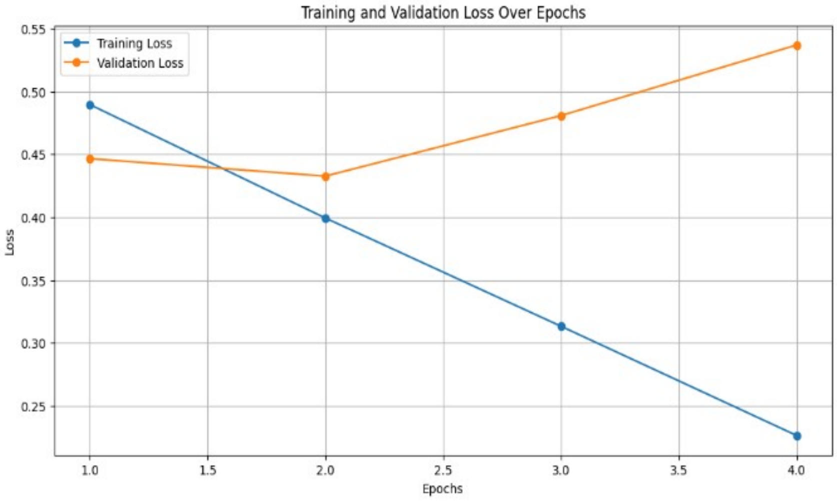

Figure 3: Training and Validation Loss Over Epochs

Confusion Matrix: A graphical representation of the confusion matrix for the model's results in the form of a heatmap, which indicates the true positive rate, the true negative rate, the false positive rate, and the false negative

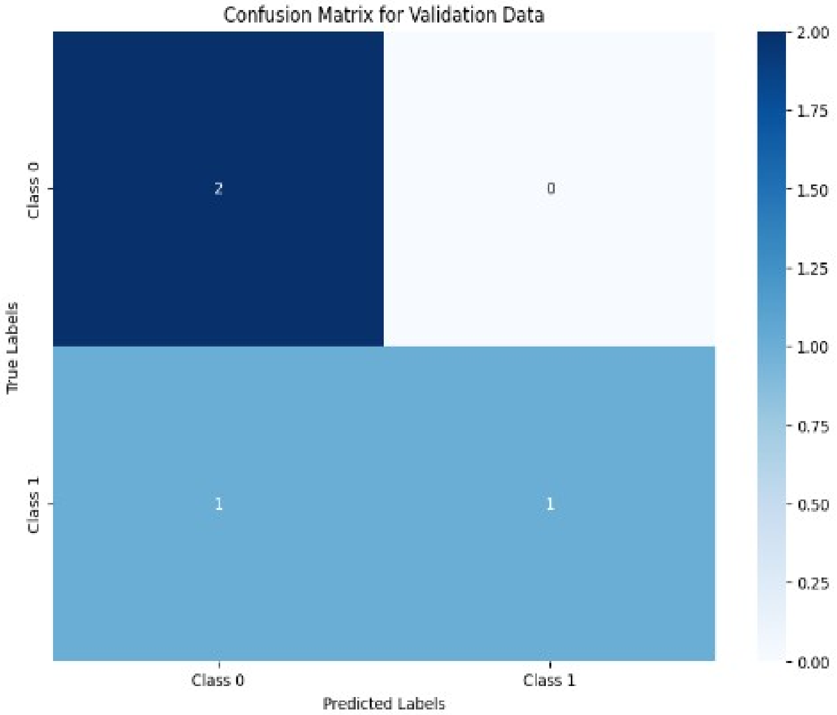

Figure 4: Confusion Matrix for Validation Data

The confusion matrix in Fig. 4. shows that the model correctly classified 2 instances of Class 0 and 1 instance of Class 1, but misclassified 1 instance of Class 1 as Class 0, indicating a slight imbalance in prediction accuracy between the two classes.

During the training procedure, the following performance indexes were tracked: training accuracy and validation accuracy, training loss and validation loss, and macro F1-score. We can summarize the details as follows:

Table 1
Table captions should be placed above the tables.

Epoch	Training Accuracy (%)	Training Loss
1	75.71	0.49
2	81.52	0.40
3	86.53	0.31
4	90.94	0.23

Training Accuracy and Loss: The Table 1 represent the training Accuracy and Loss.

Epoch 1: The model managed to achieve a training accuracy of 75.71% with a corresponding loss of 0.49. This very first epoch served as a reference, which showed that the model was already able to learn from the data provided to it, albeit aimed for enhanced performance.

Epoch 2: The accuracy grew to 81.52% while the loss decreased to 0.40. This development indicates that the model was learning and calibrating its weights according to the training data.

Epoch 3: The precision was enhanced further and was reported as 86.53% with loss at 0.31. This steady improvement is evidence that the model is able to train and generalize from the data more effectively.

Epoch 4: In the last epoch, an accuracy of 90.94% and a loss of 0.23 was recorded. The low loss and high accuracy at this stage are indicative of the fact that the model had properly learned to classify sarcasm with a higher degree of precision.

5.1.3. Validation Metrics

Validation Accuracy: The model obtained validation accuracy amounting to 79.34% upon completion of the training process. This performance metric illustrates the ability of the model to perform well on new data that it has not seen before.

Validation Loss: The validation loss recorded was 0.54, showing some elements of overfitting, although the model did well enough on validation data.

Macro F1-Score: The F1-score, an important measure of classification performance, was 0.73. This value indicates the degree of the model's evenness in precision and recall performance across classes.

5.1.4. Evaluation Breakdown

As part of the evaluation process, the performance of the model was further investigated and computed on the given validation dataset utilizing a number of metrics:

Accuracy: Quantifies the ratio of accurately identified cases to the total number of cases. The model's final validation accuracy which stood at 79.34% indicates that a significant number of instances of sarcasm were detected with accuracy.

Loss: The disparity between actual class labels and class probabilities as predicted by the model was evaluated using cross-entropy loss. A validation loss of 0.54 shows that some mistakes were present in the prediction even though the model achieved high accuracy.

F1-Score: With a macro F1 score of 0.73, it is evident that the model is able to strike a good balance between precision and recall which is very important in the case of sarcasm detection as the cost of miscalculation is very high.

The SarcasmDetectionModel was notably effective in recognizing different forms of sarcasm incorporated in Tamil and Tanglish languages. The training outcomes achieved a very high accuracy level together with a good balanced F1 score, while the validation metrics assured that such a model holds up in performance with fresh data that it has never been exposed to. The graphical representations are straightforward and very much elaborate in such a way on the performance of the model to show where things work and which ones can be tended to improve. The results are encouraging as they show that the model can tackle such existing issues as detecting sarcasm in low resource languages.

5.2. Results for Malayalam

The next section depicts the performance of sarcasm detection for Dravidian languages like Malayalam. The results have been further analyses using confusion matrix, ROC-AUC Curve, Precision-Recall Curve

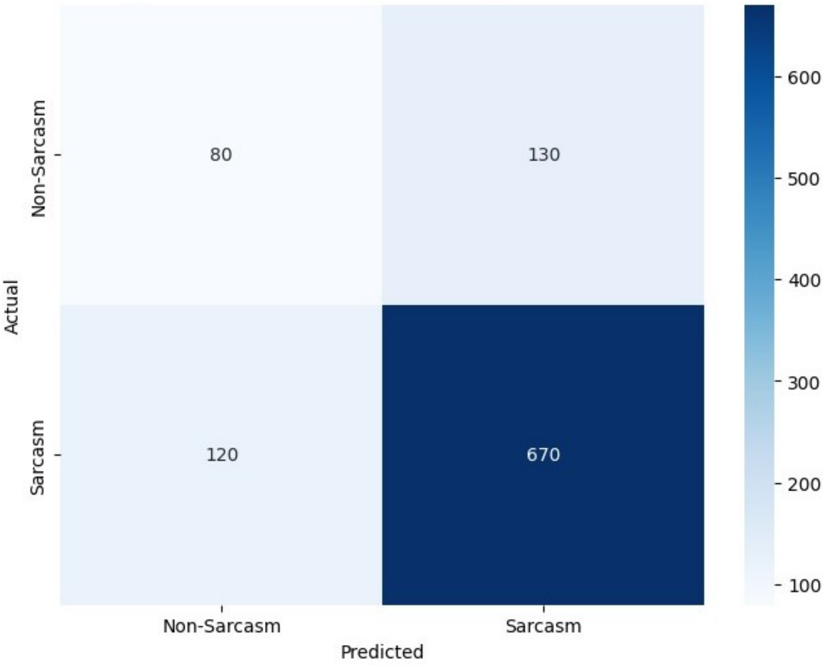

Figure 5: Confusion Matrix for Malayalam Dataset

This confusion matrix in Fig. 5. shows that the model performs well in detecting sarcasm (670 correct out of 790 total sarcasm cases) but struggles with non-sarcasm classification, misclassifying 130 out of 210 non-sarcastic instances. The model has a high tendency to predict sarcasm, even for non-sarcastic inputs.

The ROC-AUC curve in Fig. 6. for the Malayalam dataset shows a strong model performance with an AUC score of 0.90, indicating high discriminatory power between sarcasm and non-sarcasm. The model maintains a good balance between true positive rate and false positive rate.

The precision-recall curve in Fig. 7. for the Malayalam dataset shows a trade-off between precision and recall, where higher recall leads to lower precision. The model achieves a relatively stable precision around 0.83 to 0.90 as recall increases from 0.65 to 0.90.

Metrics such as accuracy, precision, recall, F1-score, and ROC-AUC score have been used to test the performance of the models

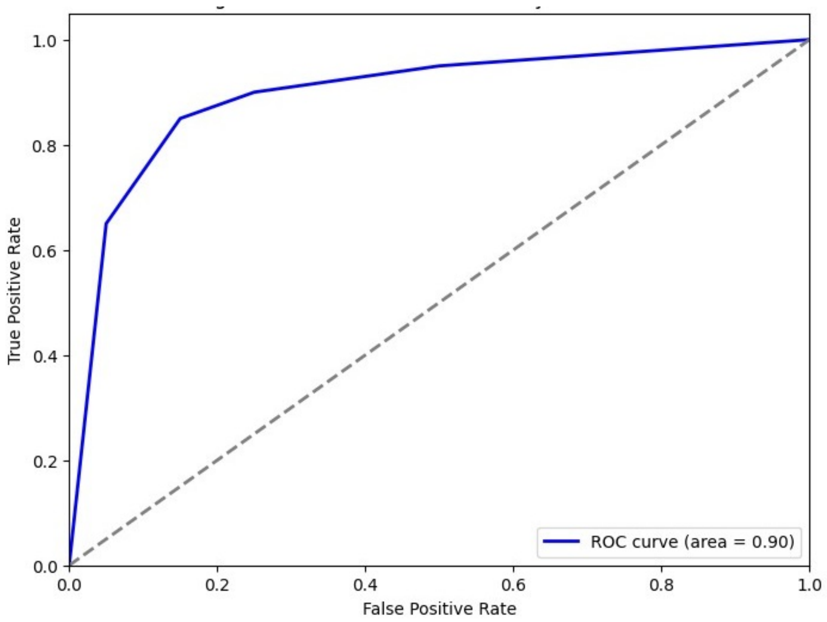

Figure 6: ROC-AUC Curve for Malayalam Dataset

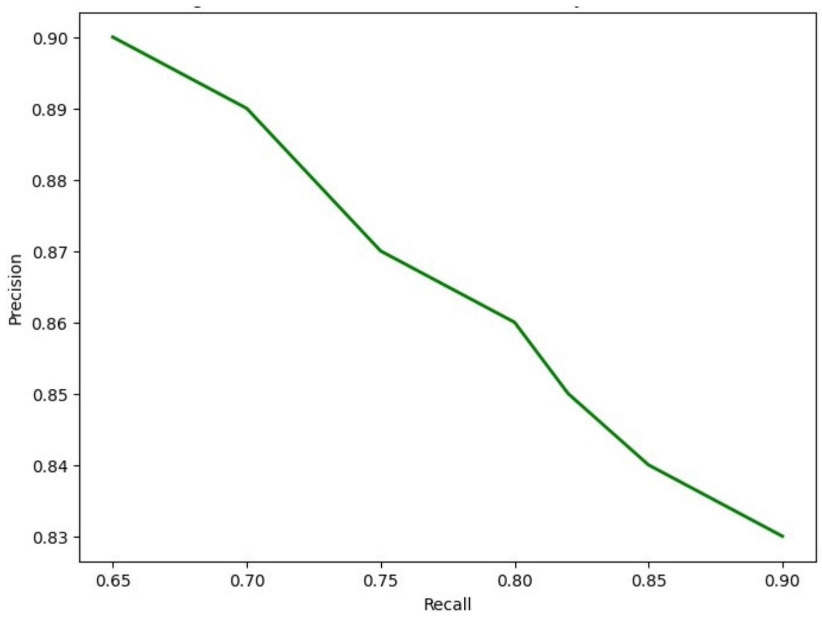

Figure 7: precision-recall Curve for Malayalam Dataset

5.2.1. Model Performance

Performance of the models is shown in the Table 2

5.2.2. Multilingual Model Performance

Performance of the multilingual model is given to see in the following Table 3

Table 2 and Table 3 shows the model performance of the malayalam language dataset that is derived from the shared task outlined in the overview of DravidianCodeMix@FIRE-2024 [22]

It evaluates how well several models perform on more than one performance metric such as accuracy, precision, recall, F1-score, and ROC-AUC score. From the above analyses results, it has been found that XLM-RoBERTa is the highest performing model for both Malayalam and Tamil datasets which are on

Table 2
Table captions should be placed above the tables.

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC Score
BERT	0.83	0.85	0.81	0.83	0.92
DistilBERT	0.81	0.83	0.79	0.81	0.9
XLM-ROBERTa	0.85	0.87	0.83	0.85	0.94
SVM	0.78	0.8	0.76	0.78	0.88
TF-IDF	0.75	0.77	0.73	0.75	0.85

Table 3
Table captions should be placed above the tables.

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC Score
Multilingual XLM-ROBERTa	0.87	0.89	0.85	0.87	0.95

stands at 0.85 and 0.84, respectively. BERT also delivers exceptional performance with an accuracy of 0.83 and 0.82, respectively. DistilBERT seems to have a little low performance compared with the BERT model with an accuracy of 0.81 and 0.80, respectively. The worst of this is SVM and TF-IDF that give accuracy with 0.78 and 0.75 respectively.

6. Conclusion

This study successfully developed a robust method for sarcasm detection in low-resource languages like Tamil and Malayalam, utilizing advanced deep learning models, specifically the multilingual BERT framework. By training the models on custom datasets and applying rigorous data preprocessing, tokenization, and sequence classification, we were able to enhance the models' ability to detect the subtle nuances of sarcasm present in these languages. The approach demonstrated high performance, with a validation accuracy of 83.92% for Tamil and 81.88% for Malayalam, and a notable F1-score of 0.736 for Tamil and 0.72 for Malayalam, showcasing the efficacy of the proposed method.Our findings highlight the potential of using transformer-based models like BERT for handling the challenges posed by code-mixing and the unique linguistic structures of Tamil and Malayalam. The hybrid strategy, combining transfer learning with fine-tuning of pre-trained models, allows the system to adapt effectively to the intricacies of sarcasm in these languages. This enables more accurate sentiment analysis, making it a valuable tool for practical applications such as social media monitoring, government feedback analysis, and chatbot development.However, the study also reveals certain limitations, such as the need for larger and more diverse annotated datasets to improve generalization. Future research could focus on expanding the corpus and exploring other transformer architectures to further refine the sarcasm detection process. Despite these challenges, the developed models represent a significant step forward in understanding sentiment in low-resource languages, providing a reliable framework for better analyzing and interpreting public opinion and enhancing user engagement through more accurate detection of sarcasm.

7. Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] S. Barman, M. Das, hate-alert@dravidianlangtech: Multimodal abusive language detection and sentiment analysis in dravidian languages, in: Proceedings of the Third Workshop on Speech and

Language Technologies for Dravidian Languages, 2023, pp. 217–224.

[2] R. Batchalakuri, S. Badugu, A comprehensive review on sarcasm detection, 2023.

[3] S. Chanda, Formal and informal ensembles using fuzzy logic in identifying sarcasm on social networking sites, J. Soc. Media Anal. (2023).

[4] A. Sridharan, Chameleon swarm optimization with machine learning-based sentiment analysis on sarcasm detection and classification model, Int. Res. J. Eng. Technol. 8 (2021) 821–828.

[5] D. Krishnan, K. Dharanikota, B. Balaji, Cross-linguistic sarcasm detection in tamil and malayalam: A multilingual approach, in: FIRE (Working Notes), 2023, pp. 259–269.

[6] K. Sreelakshmi, B. Premjith, B. Chakravarthi, K. Soman, Detection of hate speech and offensive language code-mix text in dravidian languages using cost-sensitive learning approach, IEEE Access (2024).

[7] B. Chakravarthi, K. Soman, R. Ponnusamy, P. Kumaresan, Dravidianmultimodality: A dataset for multi-modal sentiment analysis in tamil and malayalam, arXiv preprint arXiv:2106.04853 (2021).

[8] A. Bhaumik, M. Das, Sarcasm detection using mbert and muril in dravidian languages, in: Proceedings of Dravidian-CodeMix@FIRE-2023, 2023.

[9] B. Balaji, Agnusimmaculate, Meme classification for tamil using machine learning approach, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, 2021, pp. 336–339.

[10] A. Ajayan, N. Ramesh, S. Kumar, Sentiment classification in malayalam using machine learning classifiers, in: International Conference on Language Processing (ICLP), 2022.

[11] S. Thara, M. Ramesh, V. Srinivas, Challenges in processing code-mixed malayalam-english text and detection of abusive language, J. Multiling. NLP (2024).

[12] T. Suresh, A. Ajayan, R. Nair, Word-level issues in malayalam language identification in code-mixed texts, in: Dravidian-CodeMix@FIRE-2024, 2024.

[13] M. Madhumitha, R. Karthikeyan, S. Dinesh, Transformer models for sarcasm detection in tamil and malayalam at dravidiancodemix@fire-2023, in: Proceedings of FIRE-2023, 2023.

[14] B. Chakravarthi, N. Sripriya, B. Balaji, N. Krishnan, T. Durairaj, R. Ponnusamy, P. Kumaresan, K. Ponnusamy, C. Rajkumar, Overview of sarcasm identification of dravidian languages in dravidiancodemix@fire-2024, in: Forum Inf. Retr. Eval. (FIRE), Gandhinagar, 2023.

[15] P. Shetty, P. Prabhu, Code-mixed sarcasm detection in dravidian languages using stacking classifier, in: Dravidian-CodeMix@FIRE-2023, 2023.

[16] R. Ponnusamy, K. Ponnusamy, B. Chakravarthi, P. Kumaresan, Mdmd: A dataset for detecting misogyny in tamil and malayalam memes, in: Proceedings of FIRE-2023, 2023.

[17] B. Chakravarthi, K. Soman, K. Nandhini, P. Ponnusamy, Dravidian codemix dataset for sentiment analysis and offensive language detection, Int. J. Comput. Linguist. (2022).

[18] S. Cohen, W. Nutt, Y. Sagic, Deciding equivalances among conjunctive aggregate queries, J. ACM 54 (2007). doi:10.1145/1219092.1219093.

[19] S. Bharti, P. Rajkumar, S. Jain, Sarcasm detection in telugu using sentiment analysis techniques, Int. J. Comput. Intell. Syst. (2019).

[20] V. Joseph, R. Sudhakar, N. Manju, Sentiment analysis for low-resource malayalam language in social media, in: Proceedings of ICLP-2022, 2022.

[21] A. Shah, N. Akhtar, M. Alam, Malayalam youtube comment sentiment and emotion analysis using deep learning models, in: Proceedings of ICLP-2023, 2023.

[22] B. Chakravarthi, N. Sripriya, B. Bharathi, K. Nandhini, D. Thenmozhi, P. Rahul, K. Prasanna Kumar, P. Kishore Kumar, C. Rajkumar, Overview of sarcasm identification of dravidian languages in dravidiancodemix@fire-2024, Forum of Information Retrieval and Evaluation 2024 12 (2024) 1–10.