

Unmasking Sarcasm: Exploring mBERT+CNN and LSTM Models for Sarcasm Identification in Code-Mixed Tamil and Malayalam Texts

Sonith D¹, Kavya G^{2,*} and H L Shashirekha³

Department of Computer Science, Mangalore University, Mangalore, Karnataka, India

Abstract

On social media platforms, users express their thoughts and emotions in diverse ways including sentiments, sarcasm, signs of depression, and expressions of hatredness. Sarcasm is a form of expression where the intended meaning is opposite to the literal meaning of the words used, often to convey mockery or irony. This can significantly alter the perceived sentiment of a message, making it challenging for Natural Language Processing (NLP) tasks such as opinion and sentiment analysis. Detecting sarcasm is crucial because it helps in accurately interpreting user sentiments and can enhance the effectiveness of automated systems in processing and responding to user-generated content. In this direction, "Sarcasm Identification of Dravidian Languages in DravidianCodeMix@FIRE-2024"- a shared task organized at Forum for Information Retrieval Evaluation (FIRE) 2024, invites the research community to address the challenges of sarcasm detection in code-mixed Dravidian languages (Tamil-English and Malayalam-English). To explore the strategies for sarcasm identification in Dravidian languages, in this paper, we - team MUCS, describe the models proposed for the shared task. Two distinct models: i) Long Short-Term Memory (LSTM) model - trained using Keras embeddings, and ii) mBERT+CNN model - a combination of Transfer Learning (TL) (fine-tuning Multilingual Bidirectional Encoder Representations from Transformers (mBERT)) and Deep Learning (DL) approach (Convolutional Neural Network (CNN)) for building classifier, are proposed for sarcasm identification. Further, to overcome the data imbalance issue in the dataset, text augmentation technique is explored using Contextual word embedding augmenter from Natural Language Processing Augmentation (NLPAug) library, to increase the number of samples in the minority class. Among the proposed models, mBERT+CNN model outperformed LSTM model with macro F1 scores of 0.74 and 0.72 for Tamil-English and Malayalam-English datasets securing 1st and 4th ranks respectively.

Keywords

Sarcasm Detection, Data Augmentation, Transfer Learning, Deep Learning

1. Introduction

Social networking sites have become a great source of user-generated textual content that lends itself to engagement via likes, shares, comments, and discussion [1, 2, 3]. This textual content corresponds to various topics such as hate speech, hope speech, fake news etc., and sarcasm is one among them. Sarcastic comments often draw a contrast between the literal words expressed and their intended implication. For instance, if someone says, "Oh, what a masterpiece!" after watching a poorly received movie, the literal praise is meant sarcastically. In contrast, non-sarcastic comments are straightforward and convey the speaker's true sentiment. For example, saying, "The movie was really engaging", directly communicates genuine praise without any hidden meaning. Sarcastic text poses a unique challenge for text analysis due to its reliance on contextual and frequently nuanced indications [4, 5]. Satirical remarks on social media have the power to skew the interpretation of a discussion or a piece of content, which can have an impact on interactions and public opinion. Accurately detecting irony can boost the effectiveness of sentiment analysis systems and improve the analysis of comments/posts collected from social media. Separating sincere statements from satirical ones is also crucial for proper interpretation and response in the context of social media monitoring and content management. Hence, detecting sarcasm on social media platforms becomes crucial. However, sarcasm detection presents significant

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

*Corresponding author.

✉ sonithksd@gmail.com (S. D); kavyamujk@gmail.com (K. G); hlsrekha@mangaloreuniversity.ac.in (H. L. Shashirekha)

🌐 <https://www.mangaloreuniversity.ac.in/shashirekha> (H. L. Shashirekha)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

challenges as sarcastic remarks often depend on contextual understanding, tone, and cultural nuances that are difficult to capture with standard text processing techniques. The code-mixed nature of textual content on social media platforms adds its share of challenges to detect sarcastic content.

Many of the works on sarcasm detection focus on high-resource languages, leaving several low-resource languages unexplored in this direction. Sarcasm detection in low-resource Dravidian languages like Tamil and Malayalam, are exacerbated by the complexity of linguistic structures, the rich variety of expressions used in text to express sarcasm [6, 7] and the unavailability of annotated data, in these languages. To address the challenges of detecting sarcastic content in code-mixed Tamil-English and Malayalam-English text, in this paper, we, team MUCS describe the learning models submitted to “Sarcasm Identification of Dravidian Languages in DravidianCodeMix@FIRE-2024” shared task¹ organized at Forum for Information Retrieval Evaluation (FIRE) 2024. Sarcasm detection is modeled as a binary classification problem of assigning ‘Non-sarcastic’ or ‘Sarcastic’ label for the given code-mixed Tamil-English and Malayalam-English text. Two distinct models: i) LSTM model trained using Keras embeddings, and ii) mBERT+CNN model - a combination of TL (fine-tuning mBERT) and DL (CNN) for building classifier, are proposed for sarcasm detection. As the datasets provided by the organizers for this task are imbalanced, text augmentation techniques are used to increase the samples of the minority class in the datasets. Text augmentation is expected to enhance the ability to identify and interpret sarcastic comments effectively. The sample text with their corresponding labels from the given datasets for code-mixed Tamil-English and Malayalam-English are shown in Table 1.

The rest of the paper is organized as follows: Section 2 describes the recent literature on the sarcasm identification in social media texts and Section 3 focuses on the description of the proposed models followed by the experiments and results in Section 4. The paper concludes with future works in Section 5.

Table 1

Samples of code-mixed Tamil-English and Malayalam-English comments along with their English translations and labels

Language	Sample Text	Translated Text	Label
Tamil-English	நாடக காதல் சும்பலை தோலுரிக்க வருகிறாள் திரௌபதி..... மோகன் G படம் வெற்றி பெற வாழ்த்துக்கள்	Thirelapathi is coming to skin the drama love gang..... Mohan G wishes the film a success	Non-sarcastic
	Very very super trailer bayangarama eruku appo movie sollava veannum	I have to tell you that the super trailer is very terrible	
	நாடக காதல் சும்பலின் கூடாரத்தில் தீ வைத்து போல் உள்ளது இந்த திரௌபதி.	This The Raupathi is like setting fire to the tent of drama love gang.	Sarcastic
	Thaaru maaru vera lvl dhanush anna acting	Dhanush Anna's acting is on a different level	
Malayalam-English	ദിലീപ്പേട്ടൻ ഉണ്ടോ പടം ഹിറ്റാണ് All kerala dileep fans hit like button	All kerala dileep fans hit like button	Non-sarcastic
	Teaser ഇറങ്ങിയല്ല അതിനു മുമ്പേ dislike അടിച്ച avaradhi makkllzz	Avaradhi makkllzz disliked the teaser before it came out	
	നമ്മുടെ അനു സിത്താര ചേച്ചി ഇങ്ങനെ വരുന്നത് നോക്കി ഇരുന്ന് നിരാശരായി മടങ്ങിയവർ ഇവിടെ come on.....	Those who came back disappointed after watching our Anu Sithara sister coming like this, come on here....	Sarcastic
	Comments വായിക്കാൻ വന്നവർ ebde commomn	Those who came to read comments ebde commomn	

2. Related Work

Sarcasm identification involves detecting ironic or mocking language, where the intended meaning is often the opposite of the literal meaning of the words used. This task is challenging due to the nuanced nature of sarcasm, which can vary greatly across different languages and contexts [8]. There have been extensive research conducted on sarcasm detection in several Indian languages. Few of the relevant works are described below:

¹<https://sites.google.com/view/dravidian-codemix-fire2024/>

For sarcasm identification in Tamil and Malayalam social media texts, Indirakanth et al. [9] proposed Support Vector Machine (SVM) trained with Term Frequency-Inverse Document Frequency (TF-IDF) of word n-grams in the range (1, 3) and TL (BERT, Distilled BERT (DistilBERT), Cross-lingual Language Model - Robustly optimized BERT approach (XLM-RoBERTa)). Among their proposed models, DistilBERT model outperformed other models with macro F1 scores of 0.68 and 0.63 for Tamil and Malayalam datasets respectively. Pandey and Singh [10] implemented various learning models (Machine Learning (ML), DL, and TL) for identifying sarcasm in the code-mixed Hindi-English text. The ML classifiers (Naive Bayes (NB), Logistic Regression (LR), K-Nearest Neighbours (K-NN), SVM, Random Forest (RF), Decision Tree (DT), Extreme Gradient Boosting (XGB)) were trained with TF-IDF of word uni-grams, DL models (Deep Neural Network (DNN) CNN, LSTM) with Keras embeddings, and TL-based models using fine-tuned BERT. Further, they implemented a hybrid model by stacking LSTM network at the final layer of BERT model. Among the proposed models, hybrid model outperformed other models with macro F1 score of 0.98 for identifying sarcasm in the code-mixed Hindi-English text. Kolhe [11] developed Marathi dataset with 2,400 Marathi tweets for binary classification, to identify sarcasm contents in tweets and used various ML models (XGB, DT, RF and SVM) trained with TF-IDF of word unigrams, to benchmark their dataset. Among the proposed ML models, XGBoost classifier trained with TF-IDF of word unigrams, outperformed other models with a macro F1 score of 0.65.

Bhaumik and Das [12] proposed TL-based models (mBERT and Multilingual Representations for Indian Languages (MuRiL)) for sarcasm detection in Tamil-English and Malayalam-English dataset. Among their proposed models MuRiL model obtained macro F1 scores of 0.781 and 0.731 for Tamil-English and Malayalam-English texts respectively. For sarcasm detection in Tamil and Malayalam Dravidian code-mixed texts, Chanda et al. [13] implemented TL-based BERT model with an additional layer of neural networks to classify comments as sarcastic or non-sarcastic and obtained macro F1 score of 0.72 for both languages. Eke et al. [14] experimented three approaches: i) a Bidirectional LSTM (BiLSTM) model using Global Vectors (GloVe) embeddings, ii) a BERT-based model, and iii) a feature fusion model incorporating BERT features, sentiment-related information, syntactic features, and GloVe embeddings, for sarcasm detection in English social media and internet texts. Among their proposed models BiLSTM and feature fusion models outperformed other models with macro F1 scores of 0.98 and 0.80 on Twitter and Internet Argument Corpus version two (IAC-v2) datasets, respectively. Kalaivani and Thenmozhi [15] implemented ML models (LR, RF, XGB, Support Vector Classifier (SVC), and Gaussian Naive Binomial (GNB)) trained with Doc2Vec and TF-IDF of word unigrams, DL model (Recurrent Neural Network with Long Short Term Memory (RNN-LSTM)) trained with Keras embeddings, and TL-based model (a pretrained BERT variants) fine-tuned with BERT features, for identifying sarcasm in English text obtained from Twitter and Reddit forums. Among their proposed models, TL-based BERT model obtained better F1 scores of 0.722 and 0.679 for the Twitter and Reddit forums respectively.

The related work illustrates that researchers have employed ML, DL, and TL models with various features for sarcasm identification in Indian languages as well as in English. However, the performance of not all models are promising. Further, the continuous evolution of user-generated content indicates substantial opportunities for further research and innovation in this domain.

3. Methodology

The proposed methodology for sarcasm identification in code-mixed Tamil-English and Malayalam-English texts includes: Text Augmentation, Pre-processing, Feature Extraction and Model Building. The framework of the proposed methodology is shown in Figure 1 and the steps are explained below:

3.1. Text Augmentation

A significant variation in the number of instances belonging to the classes in a dataset indicates data imbalance and this causes the learning models to be biased in favor of the majority class thereby performing poorly on the minority class. This imbalance can be handled either by increasing samples

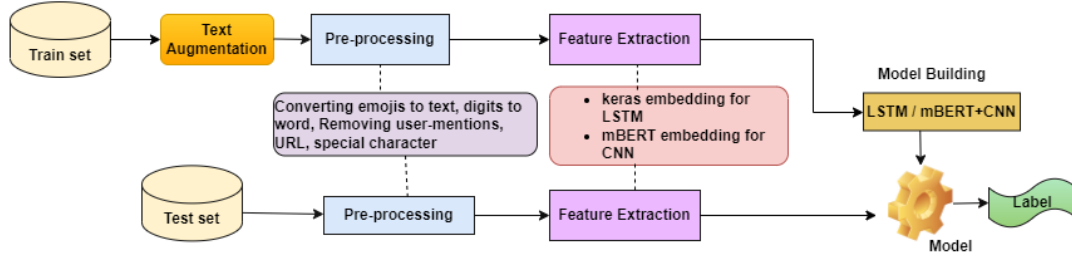


Figure 1: Framework of the proposed LSTM and mBERT+CNN models

Table 2

Statistics of code-mixed Tamil-English and Malayalam-English dataset

Label	Tamil-English			Malayalam-English		
	Train set	Dev set	Test set	Train set	Dev set	Test set
Non-sarcastic	21,740	4,630	4,621	10,689	2,305	2,314
Sarcastic	7,830	1,706	1,717	2,499	521	512

in the minority classes or decreasing samples in the majority classes. However, decreasing the samples from majority class is not a good move if the data is less in size. This leads to increasing the size of samples in minority class. The statistics of the code-mixed Tamil-English and Malayalam-English datasets provided by the shared task organizers² for sarcasm detection are shown in Table 2. From the table, it can be observed that, in contrast to the 'Non-Sarcastic' class, the number of samples belonging to 'Sarcastic' class in the Train sets of both the languages show an enormous difference and hence the dataset is unbalanced.

In order to balance the dataset, the number of samples can be expanded by either duplicating the current samples or creating new synthetic data based on the current samples [16]. While duplicating the existing samples just increase the size of the dataset, creating new synthetic samples add variation in the data in addition to increasing the size of the dataset. The literature has suggested a number of methods such as text augmentation, vector space resampling, and oversampling, to address the problem of class imbalance by boosting the samples in the minority class. Text augmentation refers to a collection of techniques used to artificially increase the size of the dataset [17]. This approach has become essential, particularly for situations where the dataset exhibits an uneven distribution of classes or a relatively low amount of labeled data. In this work, text augmentation techniques are used to boost the 'Sarcastic' (minority) class samples in the Train sets of both the languages.

Natural Language Processing Augmentation (NLPAug) - NLPAug³ is a Python library designed to facilitate text augmentation for NLP tasks. It provides a variety of transformations for text data at character-level, word-level, and sentence-level. NLPAug supports various techniques like synonym replacement, random insertion and deletion of words and characters, and shuffling of words and characters [18]. It also includes more advanced techniques/augmenters integrated with pretrained word embeddings (Word2Vec, GloVe, and BERT embeddings), contextual word embeddings, and back translation, to generate new words/text. These techniques help to enhance the diversity and robustness of the datasets, thereby improving the performance of the learning models. In this work, contextual word embedding augments (NLPAugContextualWordEmbAug) technique is used to augment the 'Sarcastic' class samples by employing 'insert' option. This technique uses multilingualBERT⁴ - a pretrained language model, to predict the words that have to be inserted at random positions in the original

²<https://codalab.lisn.upsaclay.fr/competitions/19310>

³<https://nlpaug.readthedocs.io/en/latest/>

⁴<https://huggingface.co/google-bert/bert-base-multilingual-cased>

sentence and the words are chosen automatically based on their contextual relevance by maintaining the overall meaning and coherence of the sentence. In Tamil-English and Malayalam-English datasets, sarcastic samples were increased to 21,740 and 10,689 samples respectively, to match the number of non-sarcastic samples. The examples of Tamil-English and Malayalam-English sample texts and augmented text using NLPAugContextualWordEmbAug technique are shown in Table 3.

Table 3
Sample text and augmented text in Tamil-English and Malayalam English dataset

Tamil-English	
Sample Text	Augmented Text
Avara Contol Pannunga Pls... Vera level expression... Thala...	Santo Avara Contol San Pannunga San Pls... de Vera Vera level and expression... Villa Thala...
Family sentiment... Pondatti pullainga... Engayo link aavuthe Sun pictures nalla panringa THALAIVAR DHARISANAM	Family romantic sentiment... Jaya Pondatti pullainga... Raja Engayo The link to aavuthe Sun pictures in nalla panringa... THALAIVAR IN DHARISANAM
Malayalam-English	
Jimki kammal ayi,mukkuthim ayi,aduthath enthanoo avooo..malayalikale mandanmarakkan patiila	o Jimki the kammal pala ayi, and mukkuthim pal ayi, and aduthath enthanoo ja avooo..O malayalikale... mandanmarakkan patiila
Kopile song. Lyrics etho 1st std padikuna kutty ezhuthiya pole undu. Pavam sreyaagoshal	in Kopile and song.', 'Lyrics an etho for 1st 3rd std padikuna the kutty ezhuthiya na pole undu.', 'as Pavam sreyaagoshal

3.2. Pre-processing

Raw text often contains various forms of noise and irrelevant information that can adversely affect the performance of text classification models. Pre-processing is an initial step in text classification that involves transforming raw text data into a clean and structured format suitable for analysis. It also helps to standardize and simplify the text, making it easier to extract meaningful patterns and features. In this study, pre-processing includes removing user mentions (e.g., "@username"), numbers, URLs, and HTML tags and converting emojis into their textual representations to maintain consistency and enhancing the models ability to understand the context conveyed by emojis.

3.3. Model Building

The augmented dataset is pre-processed and used to construct two distinct models: i) LSTM and ii) mBERT+CNN. These models are chosen for their proven effectiveness in capturing contextual nuances and handling sequential data, which are essential for tasks like sarcasm identification.

3.3.1. LSTM - a Deep Learning Approach

DL is a kind of ML that models and comprehends complicated patterns and representations in data by using multi-layered Artificial Neural Networks (ANNs). The steps involved in implementing this approach are given below:

- **Text and Label Fusion** - pre-processed augmented text data and the corresponding label is fused to create a combined text-label format ensuring that both pieces of information are treated as a unified input for the model during training. This enhances the model's ability to learn the relationship between text features and their respective labels, thereby improving the effectiveness of the classification process. Incorporating label as part of textual content during training allows the model to learn the nuanced relationship between textual features and their associated sentiments. While labels are usually employed only as categorical identifiers in supervised learning, integrating them with the text enables joint learning, enhancing the features captured and refining overall classification performance.
- **Feature Extraction** - is a technique that involves converting raw text into numerical representations that can be fed into ML algorithms and in this study, Keras embedding is employed to facilitate this process. The Tokenizer class from Keras is utilized to convert text data into sequences of integers based on the frequency of words, with a vocabulary size of 10,000 words

Table 4

Hyperparameter and their values used in LSTM model

Hyperparameter	Value
Max Length	100
Vocabulary Size	10,000
Embedding Dimension	128
LSTM Units	64
Dense Units	1
Activation Function	sigmoid
Loss Function	binary_crossentropy
Optimizer	adam

to maintain a reasonable model size, ensuring that the embeddings can effectively generalize to unseen data without overfitting. These sequences are then padded to ensure uniform input length for the model, specifically setting a maximum sequence length of 100. This step ensures that all text inputs have the same dimensions, which is crucial for the model to process and learn from the data effectively. Keras embeddings are language-independent and generate vectors based on vocabulary, without capturing contextual nuances. This approach supports all languages, including Tamil and Malayalam, by providing a uniform representation across diverse linguistic contexts.

- **Classifier Construction** - LSTM is a type of RNN that is capable of learning and remembering long term dependencies without encountering vanishing gradient descent or exploding problems [19]. The model architecture has an embedding layer that maps each integer-encoded word to a dense vector of 128 dimensions, capturing semantic relationships between words. This is followed by an LSTM layer, which is designed to handle sequential data and capture long-term dependencies within the text. Finally, a dense layer with a sigmoid activation function is used to produce binary classification outputs. The model is compiled with the Adam optimizer and binary cross-entropy loss function, suitable for binary classification tasks. It is then trained on the padded training data and evaluated using validation/test data to monitor its performance.

The use of LSTM enhances the model's capacity to capture contextual dependencies, which is beneficial for sarcasm identification. The hyperparameters and their values used in LSTM model is shown in Table 4.

3.3.2. mBERT+CNN model

Integration of two or more techniques allows to utilize their complementary strengths for improved classifier performance [20]. In this study, mBERT is used for feature extraction, and CNN is employed for classification and the description of these are given below:

- **Feature Extraction** - uses TL for leveraging the knowledge learnt in solving a source task to solve a target task, instead of building the model from scratch and the transformer models are fine-tuned for this purpose. mBERT is a transformer model that excels at capturing multilingual and cross-lingual features and effectively manages and integrates mixed linguistic patterns and semantic nuances through its pretrained language representations, allowing it to understand and interpret the complex code-mixed language patterns found in Tamil-English and Malayalam-English texts. These embeddings serve as rich feature representations, enabling the model to benefit from pre-learned language structures and semantics [21].
- **Classifier Construction** - in recent years, ANNs have gained a lot of attention in designing prediction models [22]. CNNs are a type of feed forward ANNs that applies convolutional filters to learn and detect local patterns in data, when combined with pooling layers, effectively capturing hierarchical features and structures for tasks such as text classification. CNN architecture is

Table 5
Hyperparameter and their values used in mBERT+CNN model

Hyperparameter	Value
bert_model_name	bert-base-multilingual-cased
num_filters	100
filter_sizes	[2, 3, 4]
dropout_rate	0.2
learning_rate	2e-5
batch_size	8
num_epochs	5
max_length	128
criterion	CrossEntropyLoss
optimizer	Adam
In Channels	128 (from BERT hidden size)
Conv1d Layer Output Channels	100
Kernel Sizes (Filter Sizes)	[2, 3, 4]
Activation Function	ReLU
Pooling Method	Max pooling

employed for sarcasm detection, leveraging pretrained mBERT embeddings as features. It utilizes mBERT embeddings having an vocabulary of size 110,000 and an embedding dimension of 768, directly bypassing the need for an additional embedding layer, as these embeddings already encapsulate rich text features. CNN applies convolutional layers with 100 filters and kernel sizes of 2, 3, and 4 to capture local features from the mBERT embeddings. Each convolutional layer is followed by a ReLU activation function and max pooling to reduce dimensionality. The pooled feature maps are concatenated into a one-dimensional vector, which is then processed through a dense layer with a softmax activation function to generate the final classification probabilities. This approach combines the contextual understanding provided by mBERT with CNNs capability to optimize the models performance in detecting sarcasm content in social media data.

The combination of mBERT for feature extraction and CNN for classification capitalizes on the strengths of models, facilitating the processing of complex multilingual data while maintaining efficiency and accuracy in sarcasm identification task. The hyperparameters and their values used in mBERT+CNN model are shown in Table 5.

4. Experiments and Results

The datasets provided by the organizers of the shared task consists of YouTube comments for both code-mixed Tamil-English and Malayalam-English texts in native as well as Roman scripts, featuring a mix of Tamil or Malayalam text with English. The performances of the proposed models with and without augmentation on the Development and Test sets are shown in Tables 6 and 7 respectively. The results in these tables reveal that the use of text augmentation has significantly improved the performances of both the models. For Tamil-English Test set, LSTM models shows a notable increase in macro F1 scores from 0.21 (without augmentation) to 0.45 (with augmentation) while mBERT+CNN models demonstrate an enhanced performance with its macro F1 scores rising from 0.43 (without augmentation) to 0.74 (with augmentation). This indicates that text augmentation aids in enhancing model performance for sarcasm detection. For Malayalam-English Test set, while the macro F1 scores of LSTM models have increased from 0.15 (without augmentation) to 0.19 (with augmentation), mBERT+CNN models has maintained a consistent macro F1 score of 0.72 (for both with and without augmentation), indicating that augmentation had minimal effect on their performances. Further, from Table 7 it is clear that mBERT+CNN models obtained better macro F1 scores of 0.74 and 0.72 for code-mixed Tamil-English and Malayalam-English texts respectively. The comparison of macro F1 scores of all the participating

Table 6

Performances of the proposed models without and with augmentation on Tamil-English and Malayalam-English Development sets

Models	Languages											
	Tamil-English						Malayalam-English					
	Without Augmentation			With Augmentation			Without Augmentation			With Augmentation		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
LSTM	0.47	0.50	0.21	0.63	0.50	0.21	0.59	0.50	0.16	0.59	0.50	0.16
mBERT+CNN	0.75	0.69	0.71	0.75	0.72	0.73	0.74	0.69	0.71	0.79	0.68	0.73
P: Precision; R: Recall; F1: Macro F1 score												

Table 7

Performances of the proposed models without and with augmentation on Tamil-English and Malayalam-English Test sets

Models	Languages											
	Tamil-English						Malayalam-English					
	Without Augmentation			With Augmentation			Without Augmentation			With Augmentation		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
LSTM	0.14	0.50	0.21	0.60	0.59	0.45	0.59	0.50	0.15	0.55	0.51	0.19
mBERT+CNN	0.59	0.50	0.43	0.73	0.76	0.74	0.71	0.73	0.72	0.76	0.69	0.72
P: Precision; R: Recall; F1: Macro F1 score												

teams for the sarcasm detection task in both code-mixed Tamil-English and Malayalam-English text are shown in Figure 2 (a) and 2 (b) respectively.

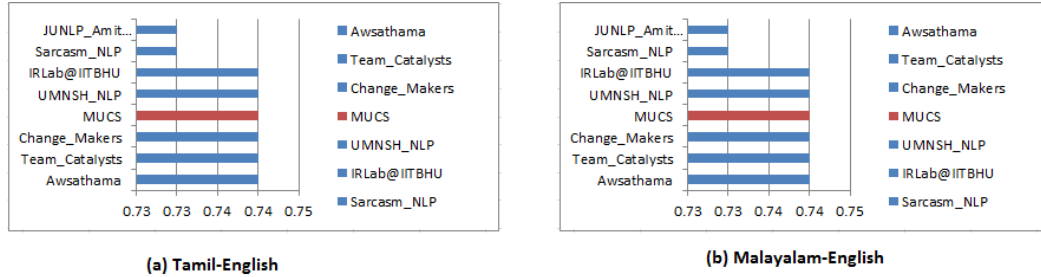


Figure 2: Comparison of macro F1 scores of the participating teams in Tamil-English and Malayalam-English dataset

Few misclassified Malayalam-English samples using mBERT+CNN model along with the actual and predicted labels and the probable reasons for misclassification are shown in Table 8.

5. Conclusion and Future Work

In this paper, we - team MUCS, describe the models submitted to 'Sarcasm Identification of Dravidian Languages in DravidianCodeMix@FIRE-2024' a shared task organized at 'FIRE 2024', to distinguish between 'Non-sarcastic' and 'Sarcastic' comments in code-mixed Tamil-English and Malayalam-English. Experiments are carried out with two proposed models: i) LSTM model - trained using Keras embeddings and ii) mBERT+CNN model - a combination of TL (fine-tuning mBERT model) and DL approach (CNN) for building classifier, for sarcasm identification. Further, to overcome the data imbalance issue in the dataset, text augmentation technique is explored using Contextual word embedding augementer from NLPAug library, to increase the number of samples in the minority class. Among the proposed models, mBERT+CNN model outperformed LSTM model with macro F1 scores of 0.74 and 0.72 for Tamil-English

Table 8
Misclassified Malayalam-English samples using mBERT+CNN model

Text	Translated Text	Actual Label	Predicted Label	Probable Reason
ഭോൾ കാരണം ഇതുവരെ കാണാതിരുന്ന പാട്ടാണു്. ഇന്നാണ് ഫസ്റ്റ് ടൈം കാണുന്നത്. എന്തിനാണാവോ ഇത്ര നല്ലൊരു പാട്ടിനെ ഡിഗ്രേഡ് ചെയ്തത്. ദാസേട്ടൻ & മമ്മൂക്ക	The song that was not seen until now because of the troll. The first time is today. Why did you degrade such a good song? Dasetan & Mammooka	Non-sarcastic	Sarcastic	The phrase "Why did you degrade such a good song?" might have been interpreted as a critical remark, and the model could have associated the context of criticism with sarcasm.
എന്താണാവോ Director ഉദ്ദേശിച്ചത്, i don't understand!	What did the director mean, i don't understand!	Non-sarcastic	Sarcastic	The comment expresses confusion, which can sometimes be interpreted as sarcasm, especially if the model has seen similar phrases used in a sarcastic context before.
Malayalathil nalla nadimar illanjittano eee ammayiyea konduvanne...nthu bore aanu avarude abhinayam porathenu enthu oooola lukkanu... sheeeeeaaaaa valare bore aayippoyi,, ee ammayi ivide kanillayirikkum ini...	Are there no good actors in Malayalam anymore? This lady's performance is so boring. Their acting is so uninteresting. She has become so boring. Maybe this lady won't be seen here anymore...	Sarcastic	Non-sarcastic	The model might have seen the repeated use of "bore" as straightforward criticism. Intense and repetitive negative language can be interpreted as Non-sarcastic rather than sarcasm.
ട്രെയിലറിന് ഒട്ടും impact ഇല്ലാലോ.. എവിടെയോ എന്തോ mistake..	If the trailer has no impact.. something is wrong somewhere..	Sarcastic	Non-sarcastic	The model might have misinterpreted "something is wrong somewhere" as a Non-sarcastic statement. Without clear sarcastic cues, it may classify it as a genuine comment questioning the trailer's impact.

and Malayalam-English datasets securing 1st and 4th ranks respectively. Advanced text augmentation techniques followed by efficient text representation methods and context-aware models to enhance sarcasm detection in social media comments will be explored further.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT in order to: Grammar and spelling check. 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, 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, DAICT , Gandhinagar, 2024.
- [2] B. R. Chakravarthi, Hope Speech Detection in YouTube Comments, in: Social Network Analysis and Mining, volume 12, Springer, 2022, p. 75.
- [3] B. R. Chakravarthi, A. Hande, R. Ponnusamy, P. K. Kumaresan, R. Priyadharshini, How Can We Detect Homophobia and Transphobia? Experiments in a Multilingual Code-mixed Setting for Social Media Governance, in: International Journal of Information Management Data Insights, volume 2, Elsevier, 2022, p. 100119.
- [4] B. R. Chakravarthi, N. Sripriya, B. Bharathi, K. Nandhini, S. Chinnaudayar Navaneethakrishnan, T. Durairaj, R. Ponnusamy, P. K. Kumaresan, K. K. Ponnusamy, C. Rajkumar, Overview of the Shared Task on Sarcasm Identification of Dravidian Languages (Malayalam and Tamil) in DravidianCodeMix, in: Forum of Information Retrieval and Evaluation FIRE - 2023, 2023.

- [5] R. Pandey, A. Kumar, J. P. Singh, S. Tripathi, A Hybrid Convolutional Neural Network for Sarcasm Detection from Multilingual Social Media Posts, in: *Multimedia Tools and Applications*, Springer, 2024, pp. 1–29.
- [6] 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 the Shared Task on Sarcasm Identification of Dravidian Languages (malayalam and tamil) in *dravidiancodemix*, in: *Forum of Information Retrieval and Evaluation FIRE-2023*, 2023.
- [7] N. Sripriya, T. Durairaj, K. Nandhini, B. Bharathi, K. K. Ponnusamy, C. Rajkumar, P. K. Kumaresan, R. Ponnusamy, C. Subalalitha, B. R. Chakravarthi, Findings of Shared Task on Sarcasm Identification in Code-Mixed Dravidian Languages, in: *FIRE 2023*, volume 16, 2023, p. 22.
- [8] R. González-Ibáñez, S. Muresan, N. Wacholder, Identifying Sarcasm in Twitter: a Closer Look, in: *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, 2011, pp. 581–586.
- [9] V. Indirakanth, D. Udayakumar, T. Durairaj, B. Bharathi, Sarcasm Identification of Dravidian Languages (Malayalam and Tamil)., in: *FIRE (Working Notes)*, 2023, pp. 327–335.
- [10] R. Pandey, J. P. Singh, BERT-LSTM Model for Sarcasm Detection in Code-mixed Social Media Post, in: *Journal of Intelligent Information Systems*, volume 60, Springer, 2023, pp. 235–254.
- [11] P. K. P. S. Kolhe, MarathiSarc: A Marathi Tweets Dataset for Automatic Sarcasm Detection of Marathi Tweets, 2022.
- [12] A. B. Bhaumik, M. Das, Sarcasm Detection in Dravidian Code-Mixed Text Using Transformer-Based Models., in: *FIRE (Working Notes)*, 2023, pp. 249–258.
- [13] S. Chanda, A. Mishra, S. Pal, Sarcasm Detection in Tamil and Malayalam Dravidian Code-Mixed Text., in: *FIRE (Working Notes)*, 2023, pp. 336–343.
- [14] C. I. Eke, A. A. Norman, L. Shuib, Context-based Feature Technique for Sarcasm Identification in Benchmark Datasets using Deep Learning and BERT Model, in: *IEEE Access*, volume 9, IEEE, 2021, pp. 48501–48518.
- [15] A. Kalaivani, D. Thenmozhi, Sarcasm Identification and Detection in Conversation Context using BERT, in: *Proceedings of the Second Workshop on Figurative Language Processing*, 2020, pp. 72–76.
- [16] L. Wang, M. Han, X. Li, N. Zhang, H. Cheng, Review of Classification Methods on Unbalanced Data sets, in: *Ieee Access*, volume 9, IEEE, 2021, pp. 64606–64628.
- [17] C. Shorten, T. M. Khoshgoftaar, B. Furht, Text Data Augmentation for Deep Learning, in: *Journal of big Data*, volume 8, Springer, 2021, p. 101.
- [18] P. Gujjar, P. K. HR, Natural Language Processing using Text Augmentation for Chatbot, in: *2022 International Conference on Artificial Intelligence and Data Engineering (AIDE)*, IEEE, 2022, pp. 248–251.
- [19] S. Hochreiter, Long Short-term Memory, in: *Neural Computation MIT-Press*, 1997.
- [20] J. Sansana, M. N. Joswiak, I. Castillo, Z. Wang, R. Rendall, L. H. Chiang, M. S. Reis, Recent Trends on Hybrid Modeling for Industry 4.0, in: *Computers & Chemical Engineering*, volume 151, Elsevier, 2021, p. 107365.
- [21] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: *CoRR*, volume abs/1810.04805, 2018. URL: <http://arxiv.org/abs/1810.04805>. *arXiv:1810.04805*.
- [22] P. Mehndiratta, S. Sachdeva, D. Soni, Detection of Sarcasm in Text Data using Deep Convolutional Neural Networks, in: *Scalable Computing: Practice and Experience*, volume 18, 2017, pp. 219–228.