# Leveraging ChatGPT and XLM-RoBERTa for Sarcasm Detection in Dravidian Code-Mixed Languages

Supriya Chanda[1], Krishna Tewari[1], Arjun Mukherjee[1] and Sukomal Pal[1]

[1]*Information Retrieval Lab (IReL), Dept. of CSE, Indian Institute of Technology (BHU) Varanasi, INDIA*

## Abstract

Sarcasm refers to a form of irony where what is meant is actually said in an opposite manner usually in a mocking or humorous form, it could be either verbal or written. We very often come across multiple sarcastic comments which are code-mixed in various social-media platforms. In order to obtain insights from the textual data available or encountered upon, we would need a system to identify the sentiments behind the text and detect sarcasm. In this paper, we present a solution submitted for the shared task titled 'Sarcasm Identification of Dravidian Languages Tamil and Malayalam, which was organized by Dravidian CodeMix 2024 at the Forum for Information Retrieval Evaluation (FIRE) 2024. This paper explores an approach to sarcasm detection, leveraging the BERT (Bidirectional Encoder Representations from Transformers) and a supplementary layer of neural networks for precise classification into two distinct classes: sarcastic and non-sarcastic comments. It also uses ChatGPT for the same and performs a comparative study between GPT and BERT-based models. Our experiment demonstrates that our model effectively detects sarcastic comments, achieving an $F_1$ score of 0.74 for both the Tamil-English and Malayalam-English code-mixed datasets, in contrary to GPT which can just achieve the $F_1$ score of 0.64 for the above mentioned datasets. This score reflects a reasonable overall performance and places us at the third position in the ranking for Malayalam-English language pairs and at the first position in the ranking for Tamil-English language pairs.

## Keywords

Social Media, Code-Mixed, BERT, ChatGPT, Sarcasm, Sentiment Analysis, Tamil, Malayalam

## 1. Introduction

In the fast pacing age of technology, we often come across certain captivating linguistic puzzles which are one of the fields humans have expertise into through tone or context and emotional cues, but for machines it remains a formidable task [1]. Among these puzzles, sarcasm remains as one of the most complicated linguistic problem. Sarcasm involves expressing thoughts in a manner that conceals the true intentions of the speaker, often infused with a dose of mockery or humor, serving as a linguistic tool to convey sentiments in a subtle manner [2].

In the domain of sentiment analysis, sarcasm plays a very significant role in understanding of data. Thus, accurately detecting sarcasm becomes a crucial aspect. In a world driven by technology, the boom in social media users has been unfolding exponentially, with a staggering 60% of the global population actively participating on these platforms, dedicating an average of 2 hours and 24 minutes daily to their online engagements (as reported by smartinsights [1]). Individuals can freely express their views across wide range of subjects, events, personalities, products and a lot more domains, generating an astounding volume of data real-time at a staggering rate of 328.77 million terabytes daily; all because of the panorama provided by social media platforms. A considerable portion of this corpus is code-mixed, i.e. a phenomenon of linguistic wherein individuals very often combine intricacies of different languages.

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

*Corresponding author.
supriyachanda.rs.cse18@itbhu.ac.in (S. Chanda); krishnatewari.rs.cse24@itbhu.ac.in (K. Tewari); arjunmukherjee.rs.cse23@itbhu.ac.in (A. Mukherjee); spal.cse@itbhu.ac.in (S. Pal)
https://supriyachanda.github.io/ (S. Chanda); https://cse-iitbhu.github.io/irlab/spal.html (S. Pal)
0000-0002-6344-8772 (S. Chanda); 0009-0005-6599-9956 (K. Tewari); 0000-0001-8743-9830 (S. Pal)



[1]https://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/

Code-mixing can very often be identified as the pinnacle of utmost experience users have of varied and diverse linguistic background. Through a peep hole of highlights we can observe the skills users possess in weaving together different languages to enhance communication, adroitly switching between tongues to convey their feelings [3]. In the globe of social media platforms, the amalgamation of multiple languages whilst online interaction, imparting comments, posts or messages; is a common occurrence and shouldn't be a matter of surprise [4]. The analysis of text not emerged in its native form presents an incremental layer of complexity in the versified area of natural language processing.

The significance of sentiment analysis application on the aforesaid data is tremendous. It not only provides valuable insights on various fields such as product description, market research, social trends and customer feedback but can also counter the spread of hate speech on social media platforms [5], thus protecting the mental well being of individuals. Henceforth, the ability to derive intelligence in user queries wrapped up with sarcasm opens up the road not taken to provide relevant information and responses to the users.

This shared task focuses on precise identification and determination of sarcasm within a code-mixed dataset comprising of comments and posts in Tamil-English and Malayalam-English. It helps to dive deeper on how sarcasm is used in mixed-language conversations sourced from social-media. Deciphering the subtleties of communication happening digitally can be achieved by gaining a deeper insight into how sarcasm works in the ever- revolving globe of online interactions.

In this paper, we applied a method that leverages XLM-Roberta and ChatGPT to enhance its capability in identifying sarcasm and determining sentiment polarity within code-mixed comments and posts written in Tamil-English and Malayalam-English, which are commonly encountered on social media platforms, and also did a comparative study between both the approaches determining which one works more efficiently on code-mixed data.

The rest of the paper is structured as follows. Section 2 provides a concise overview of prior research in this field. In Section 3, we delve into the datasets we utilized for our investigation. Section 4 elaborates on our computational methodologies, model specifications, and the techniques we employed for evaluation. Next, we present our results and conduct a comprehensive analysis in Section 5. Finally, we conclude in Section 6.

## 2. Related Work

Code-mixing in languages has been the ongoing subject of extensive research over quite a long period of time equi-validating it to around decades. However, the analysis of code-mixed text, particularly in the area of social media, represents a comparatively new frontier or domain in the field of Natural Language Processing(NLP).

In recent years, we have explored various text processing tasks on code-mixed data, focusing on different language pairs such as Bengali-English, Hindi-English, and Dravidian language pairs—specifically Tamil-English, Malayalam-English, and Kannada-English. Our research has covered word-level, sentence-level, and sentence-pair-level tasks. For instance, word-level language identification [6] is one of the foundational tasks that facilitates downstream processing of code-mixed data. We have observed that incorporating a language identification module improves the performance of sentiment analysis tasks [7, 8]. Additionally, our findings show that meta-embeddings outperform pretrained word embeddings [9]. Another key sentence-level task we addressed is hate speech identification [10]. Recently, we also explored the novel task of information retrieval on code-mixed data [11]. However, despite the advancements made, these tasks remain challenging due to the complexities inherent in processing code-mixed content.

Identification and detection of sarcasm has always been a significant task at downstream in the domain of NLP which has always attracted researchers. Multiple efforts and contributions have been made to solve this challenging problem and one such notable approach in this regard is the use of IndicBERT for detecting sarcasm in social media text, as proposed by Amir et al [12]. This model focuses on capturing contextual information and identifying sarcasm. Continuing further Wicana et al. [13] dived into different machine learning methodologies to detect sarcasm. Their work offered a wide perspective on the latest techniques and associated difficulties in deploying that model to identify and detect sarcasm. They explored a range of neural network-based classification structures, including models like subword-level LSTM, Hierarchical LSTM, BERT, mBERT, XLM RoBERTa, LSTM, GRU, and XLNet.

Without a slightest doubt IndicBERT has proven itself to be very efficient in understanding nuances and language-specific characteristics of Indian languages [14]. In order to deal with different intertwined threads if sarcasm detection, we can see researchers employing a variety of techniques. For instance, an attention-based BiLSTM model, combined with a feature-rich Convolutional Neural Network (CNN) approach [15], has been utilized. It is essential to note that while sarcasm and hate speech are related, they are not the same, and they demand distinct approaches.

Efforts to identify sarcasm in social media content have lead to pathway of various innovative approaches thus far. One such approach is employing prompt-based ChatGPT and its comparison in multilingual sarcasm detection [16]. Additionally, Hegde et al. [17] investigated using the same for Tamil and Malayalam code-mixed texts. While other approach in this domain is the use of the multilingual XLM-RoBERTa with CNN and BiLSTM for sarcasm detection [18]. Furthermore, Agrawal et al. [19] swam through the pool of emotional transitions to improve sarcasm detection, re-iterating the dynamic nature of emotional signals in detection of sarcasm. As sarcasm goes beyond just written text. Pandey and Vishwakarma [20] managed to handle the challenge of multi-modal sarcasm detection in videos. They dealt with deep learning approaches to effectively use sensory inputs, visuals and audio for sarcasm detection and identification. The organizers held a similar task last year, where we used the mBERT model and achieved good results for both language pairs [3]. From previous years findings we can see Transformer-based language models have been pivotal in advancing language comprehension [21, 22].

These studies provide a glimpse into how evolving is the area of sarcasm detection and how a wide spectrum of approaches and techniques are being applied, each offering significant insights, advancements and contribution in this field.

# 3. Dataset

The dataset that has been provided by the organizers [23] have served as a valuable resource for our research, having comprised of several code mixed comments in Tamil-English and Malayalam-English, sourced from social media. We often find that comments or post are of multiple sentences, but the dataset mainly has an average sentence length of one. The realm of dataset encouraged us to investigate how sarcasm is manifested in code-mixed contexts in social media. It includes development, training and test datasets of comments in Tamil-English and Malayalam-English, having various code-mixed and linguistic characteristics which provide a rich foundation for our research in detection and identification of sarcasm. Table 1 provides a summary of the dataset statistics for both language pairs. It includes details such as the total number of samples, the distribution across different classes or labels, and the proportion of each language in the code-mixed data. These statistics give an overall view of the dataset composition, helping to understand the balance and diversity within the dataset. Table 2 presents example sentences from the dataset for both language pairs. The table includes the original sentence, and any associated labels or annotations (such as Sarcastic or Non-sarcastic) to demonstrate the variety of samples within the dataset.

# 4. Methodology

## 4.1. Preprocessing

In the data preprocessing phase, we ensured to conduct several essential text preprocessing steps to refine the dataset. We targeted in removing hashtags, punctuation marks, URLs, numbers and

**Table 1**
Dataset Statistics and Class Distribution Across Training, Development, and Testing Splits for Tamil and Malayalam

| TAMIL-ENGLISH | | | |
|---|---|---|---|
| **Class** | **Training** | **Development** | **Test** |
| Sarcastic | 7830 | 1706 | 1717 |
| Non-Sarcastic | 21740 | 4630 | 4621 |
| Total | 29570 | 6336 | 6338 |
| MALAYALAM-ENGLISH | | | |
| **Class** | **Training** | **Development** | **Test** |
| Sarcastic | 2499 | 521 | 512 |
| Non-Sarcastic | 10689 | 2305 | 2314 |
| Total | 13188 | 2826 | 2826 |

**Table 2**
Sample Text and Corresponding Sarcasm Labels for Tamil and Malayalam Code-Mixed Data

| Text | Label |
|---|---|
| TAMIL - ENGLISH | |
| இந்த படம் திரைக்கு வர இறைவனை வேண்டுகிறேன். | Non-sarcastic |
| ரஜினி 96'ல வழுக்கையும் வெள்ளை தாடியோடயுமா இருந்தாரு.. | Sarcastic |
| MALAYALAM - ENGLISH | |
| കൊട്ടാരക്കര യിൽ നിന്നും ഏട്ടന്റെ അനിയന്മാരുടെ വക ഇരിക്കട്ടെ | Non-sarcastic |
| വന്നു വന്നു വരികൾവരെ ദരിദ്രമായി പോയി മലയാള സിനിമയുടെ ഒരു അവസ്ഥയെ. | Sarcastic |

mentions that does not have a great significance. Emojis were removed with their corresponding text representations. Also, any extra spaces were removed to ensure a clean and consistent text corpus for analysis. All our experiments were conducted after performing thorough preprocessing on the code-mixed datasets.

## 4.2. Model Architecture

In our research, we have used the bert-based-multilingual (XLM-RoBERTa) pre-trained model with some fine tuning and ChatGPT to create a basic foundation of our task. In the recent trends, we have found ChatGPT providing a tremendous boom in linguistics. So we have also employed ChatGPT for our task of sarcasm prediction in code-mixed data, we utilized the model to predict labels on the test dataset. A prompt-based approach was employed, where each test sentence was presented to ChatGPT with a carefully crafted prompt instructing the model to classify the sentence as either "Sarcastic" or "Non-sarcastic." The prompt was designed to ensure that ChatGPT understood the task in the context of code-mixed language data, taking into account the complexities of identifying sarcasm in multilingual social media content. XLM-RoBERTa is built on transformer architecture, which involves self-attention mechanism in both ends of encoding and decoding. These models are pre-trained on vast multilingual text corpora and have a track-record of delivering exceptional performance when fine-tuned with respect to the assigned tasks for downstream. For our objective of identifying sarcasm in code-mixed language, we opted for BERT (Bidirectional Encoder Representations from Transformers) model, with a focus in multilingual variant known as XLM-RoBERTa trained in 100 different languages.

We have submitted two approaches prediction files. In the first approach, we used prompt-based ChatGPT on the test dataset to classify them into Sarcastic or Not-Sarcastic. In the second approach, we used XLM-RobERTa where we took a special token as input which propagated through the layers, applying self-attention mechanisms and forwarding the output to the next consequent layer (see Figure

1). The output from final layer was fed into Neural Network which classified the component into two categories: Sarcastic or Not-Sarcastic.

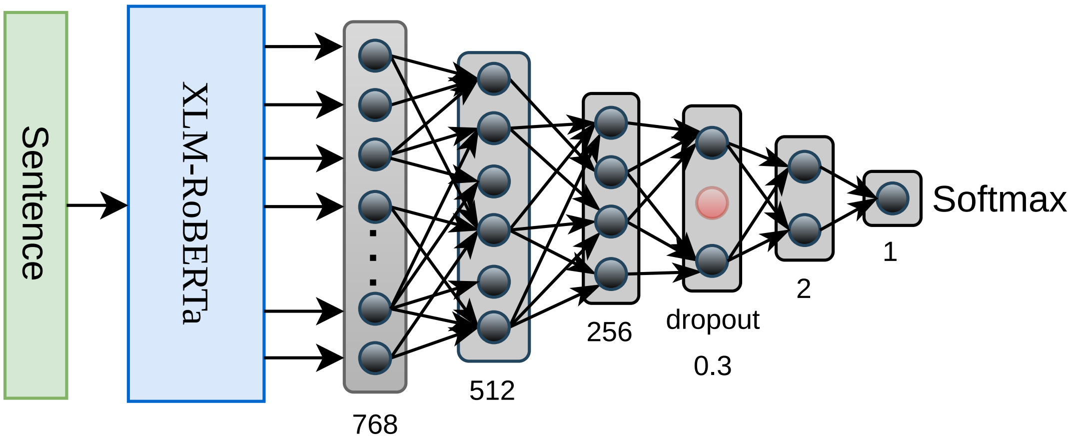



**Figure 1:** Proposed Architecture diagram

In the training phase, we carefully tuned specific hyperparameters to guide the learning process effectively. These hyperparameters were optimized to ensure the model's proficiency in sarcasm detection. We used a batch size of 16, a learning rate of 2e-5, and the AdamW optimizer to manage the weight updates. The model was trained for a maximum of 2 epochs, as we observed overfitting beyond this point using an early stopping method. This early stopping mechanism monitored the validation loss and prompted the model to exit training after 2 epochs when overfitting became apparent. The loss function employed was binary cross-entropy, given the binary nature of the sarcasm classification task. These hyperparameter settings allowed the models to achieve optimal performance without overfitting, ensuring accurate detection of sarcasm in code-mixed data.

## 5. Results and Discussion

This section is meant to deep dive into comprehensive evaluation of ChatGPT's and our proposed models performance on both datasets: Tamil-English and Malayalam-English, as part of Sarcasm Detection and Identification in code-mixed Dravidian Languages. The performance of our proposed models is examined using a range of evaluation metrics, with a primary focus on macro-averaged $F_1$-score, accuracy, recall and weighted average $F_1$-score. The test data provided to us by the organizers served as the foundation of our model evaluation.

Our methodology involved fine-tuning our model based on the training and validation datasets, ensuring it was well-prepared for the subsequent test data. Upon submission of our prediction file, for our first approach which included employing ChatGPT, we achieved an $F_1$ Score of 0.64 for both language pairs and for our second approach which involved XLM-RoBERTa we achieved an $F_1$ Score of 0.74 again for both language pairs. This score reflects a reasonable overall performance and places us at the third position in the ranking for Malayalam-English language pairs and at the first position in the ranking for Tamil-English language pairs. Table 3 and 4 display the performance of the test outcomes for our proposed model and top scored team for Malayalam-English and Tamil-English language respectively [24]. Table 5 and 6 show the class wise classification report for both language pairs on test data.

While our system demonstrated commendable accuracy, it's worth noting that other competing teams surpassed us in both Precision and Recall, which ultimately influenced our $F_1$ score and final ranking especially in Malayalam-English, also, our model could reach this F1 score in 2 epochs. This outcome encourages further refinement of our approach to enhance our model's precision and recall, aiming for even more competitive results in future endeavors. Figure 2 display the confusion matrices

**Table 3**
Participant $F_1$-scores and Ranks for Tamil-English test data (Our Score Highlighted in Bold)

| Tamil - English | | | | |
|---|---|---|---|---|
| S. No | Team Name | Runs | macro $F_1$ score | Rank |
| 1 | Awsathama | 1 | 0.74 | 1 |
| 2 | Team_Catalysts | 3 | 0.74 | 1 |
| 3 | Change_Makers | - | 0.74 | 1 |
| 4 | MUCS | 2 | 0.74 | 1 |
| 5 | UMNSH_NLP | 3 | 0.74 | 1 |
| **6** | **IRLab@IITBHU** | **2** | **0.74** | **1** |
| 7 | Sarcasm_NLP | 3 | 0.73 | 2 |
| 8 | JUNLP_Amit Barman | 3 | 0.73 | 2 |
| 9 | PixelPhrase | 1 | 0.73 | 2 |
| 10 | Codespark | 2 | 0.72 | 3 |
| 11 | Tr4nslate | - | 0.71 | 4 |
| 12 | Tech_Army_KEC | 3 | 0.70 | 5 |
| 13 | Beyond_Tech | - | 0.70 | 5 |
| 14 | SSN_Language | 1 | 0.70 | 5 |
| 15 | Code Crafters | - | 0.69 | 6 |
| 16 | CJM | - | 0.68 | 7 |
| 17 | MSD | - | 0.68 | 7 |
| 18 | The_Three_Mustketeers | 2 | 0.68 | 7 |
| 19 | KEC_AI_InnovationEngineers | 3 | 0.67 | 8 |
| 20 | KEC_AIDS_79114 | 3 | 0.61 | 9 |
| 21 | Te4Titans | 1 | 0.61 | 9 |
| 22 | Tech_Chasers | - | 0.55 | 10 |
| 23 | DLRG | - | 0.49 | 11 |
| 24 | JUNLP | - | 0.47 | 12 |
| 25 | SSNites | - | 0.24 | 13 |

**Table 4**
Participant $F_1$-scores and Ranks for Malayalam-English test data (Our Score Highlighted in Bold)

| Malayalam - English | | | | |
|---|---|---|---|---|
| S. No | Team Name | Runs | macro $F_1$ score | Rank |
| 1 | UMSNH_NLP | 1 | 0.76 | 1 |
| 2 | Awsathama | 1 | 0.75 | 2 |
| 3 | Codespark | 2 | 0.74 | 3 |
| **4** | **IRLab@IITBHU** | **2** | **0.74** | **3** |
| 5 | Sarcasm_NLP | 2 | 0.72 | 4 |
| 6 | MUCS | 2 | 0.72 | 4 |
| 7 | PixelPhrase | 2 | 0.72 | 4 |
| 8 | JUNLP_Amit Barman | - | 0.72 | 4 |
| 9 | MSD | - | 0.71 | 5 |
| 10 | CJM | - | 0.70 | 6 |
| 11 | KEC_Tech_Titan | 1 | 0.70 | 6 |
| 12 | Beyond_tech | - | 0.67 | 7 |
| 13 | Tr4nslate | - | 0.67 | 7 |
| 14 | Tech_Army_KEC | 3 | 0.67 | 7 |
| 15 | JUNLP | - | 0.66 | 8 |
| 16 | The_Three_Mustketeers | - | 0.66 | 8 |
| 17 | SSN_Language | 1 | 0.62 | 9 |
| 18 | KEC_AIDS_79114_VarshiniS H | 3 | 0.58 | 10 |
| 19 | SSNites | - | 0.57 | 11 |
| 20 | Tech_Chasers | - | 0.56 | 12 |
| 21 | TextTitans | 1 | 0.50 | 13 |

**Table 5**
Classification Report for Sarcasm Detection on Tamil-English Test Data Using ChatGPT and XLM-RoBERTa

| | **ChatGPT** | | | **XLM-RoBERTa** | | | |
|---|---|---|---|---|---|---|---|
| | Precision | Recall | $F_1$-score | Precision | Recall | $F_1$-score | support |
| Non-sarcastic | 0.81 | 0.80 | 0.80 | 0.85 | 0.89 | 0.87 | 4621 |
| Sarcastic | 0.48 | 0.49 | 0.48 | 0.65 | 0.57 | 0.61 | 1717 |
| macro avg | 0.64 | 0.65 | 0.65 | 0.75 | 0.73 | 0.74 | 6338 |
| weighted avg | 0.72 | 0.72 | 0.72 | 0.80 | 0.80 | 0.80 | 6338 |
| Accuracy | | 0.72 | | | 0.80 | | 6338 |

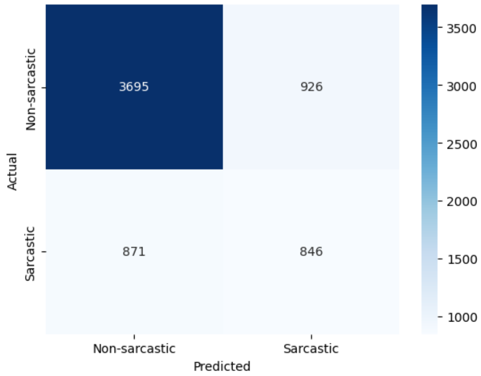
**Table 6**
Classification Report for Sarcasm Detection on Malayalam-English Test Data Using ChatGPT and XLM-RoBERTa

| | **ChatGPT** | | | **XLM-RoBERTa** | | | |
|---|---|---|---|---|---|---|---|
| | Precision | Recall | $F_1$-score | Precision | Recall | $F_1$-score | support |
| Non-sarcastic | 0.87 | 0.83 | 0.85 | 0.91 | 0.89 | 0.90 | 2314 |
| Sarcastic | 0.36 | 0.44 | 0.40 | 0.54 | 0.61 | 0.58 | 512 |
| macro avg | 0.62 | 0.63 | 0.62 | 0.73 | 0.75 | 0.74 | 2826 |
| weighted avg | 0.78 | 0.76 | 0.77 | 0.85 | 0.84 | 0.84 | 2826 |
| Accuracy | | 0.76 | | | 0.84 | | 2826 |

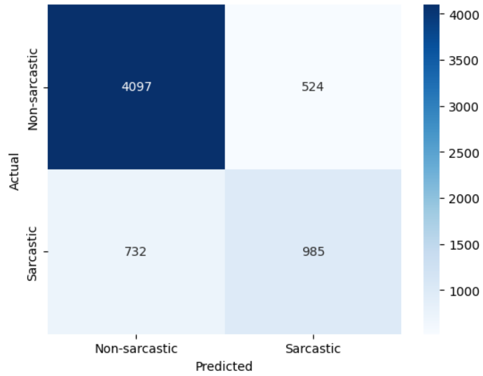
of two language pairs based on our submission. The confusion matrices indicate that XLM-RoBERTa consistently outperforms ChatGPT across both language pairs. For the Tamil-English dataset, XLM-RoBERTa achieved higher accuracy in classifying both non-sarcastic and sarcastic instances, with a significantly lower number of false negatives (524 vs. 926 for ChatGPT) and better sarcasm detection (985 true positives vs. 846 for ChatGPT). Similarly, in the Malayalam-English dataset, XLM-RoBERTa demonstrated stronger performance, with fewer misclassifications in both categories and a notably lower false-negative rate (256 vs. 400 for ChatGPT). The performance gap is more pronounced in the Malayalam-English pair, where ChatGPT struggles to detect sarcasm, achieving only 225 true positives compared to 312 for XLM-RoBERTa. These results suggest that XLM-RoBERTa is more effective in capturing the nuances of code-mixed sarcasm, providing better generalization and robustness across diverse language pairs.

## 6. Conclusion

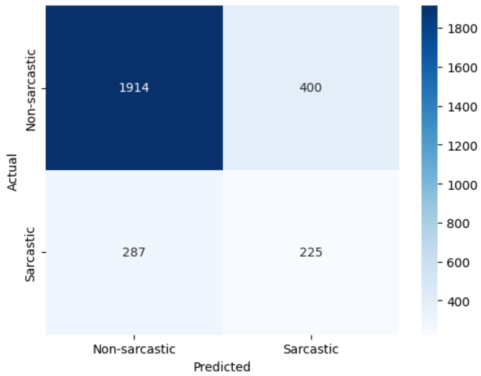
In this research, we have tried to solve the intricate task of identifying sarcasm in code-mixed comments and/or posts, especially in Tamil-English and Malayalam-English languages, extracted from the world of social media. Our dive into the waves of sentiment analysis definitely ensures us of the growing significance of user direct or indirect opinions/expressions in the context of enhancing strategies, be it business, marketing, government or other. In our experimentation, we tried to harness the power of ChatGPT, which yielded a $F_1$ score of 0.64 and the pre-trained multilingual BERT model, which produced an outstanding $F_1$ score of 0.74. This achievement shined the limelight on our approach in capturing the nuances of sarcasm in code-mixed data. Despite of impressive accuracy, we surely acknowledge the competitive environment where other teams excelled in Precision and Recall, affecting our $F_1$ score and final ranking, especially in Malayalam-English. Gaining insights from this, we are all geared up for refining our methodology further for both our approaches, especially fine-tuning the approach where we use ChatGPT and steadfast in our mission of enhancing precision and recall, with the aim of achieving more competitive results.



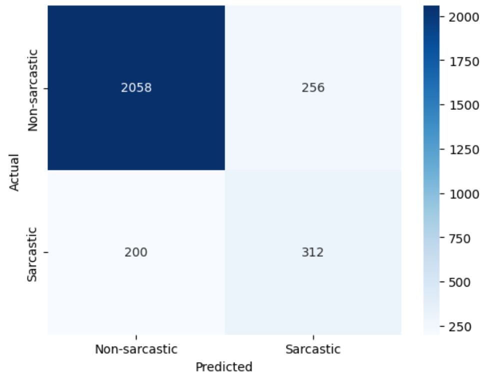

(a) ChatGPT for Tamil-English




(b) XLM-RoBERTa for Tamil-English




(c) ChatGPT for Malayalam-English




(d) XLM-RoBERTa for Malayalam-English

**Figure 2:** Confusion Matrices for Sarcasm Detection on Tamil-English and Malayalam-English Test Data Using ChatGPT and XLM-RoBERTa

## Declaration on Generative AI

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

## References


[1] B. Liu, Sentiment analysis and subjectivity, in: Handbook of Natural Language Processing, Second Edition, Chapman and Hall/CRC, 2010, pp. 627–666. URL: http://www.crcnetbase.com/doi/abs/10.1201/9781420085938-c26.

[2] P. B. Aditya Joshi, M. J. Carman, Automatic sarcasm detection: A survey, ACM Comput. Surv. 50 (2017) 73:1–73:22. URL: https://doi.org/10.1145/3124420.

[3] S. Chanda, A. Misha, S. Pal, Sarcasm detection in tamil and malayalam dravidian code-mixed text., in: FIRE (Working Notes), 2023, pp. 336–343.

[4] J. W. Utsab Barman, Amitava Das, J. Foster, Code mixing: A challenge for language identification in the language of social media, in: Proceedings of the First Workshop on Computational Approaches to Code Switching@EMNLP 2014, Doha, Qatar, October 25, 2014, Association for Computational Linguistics, 2014, pp. 13–23. URL: https://doi.org/10.3115/v1/W14-3902.

[5] S. Chanda, S. Pal, Hate Content Identification in Code-mixed Social Media Data, Chapman and Hall/CRC, 2025, pp. 225–247.

[6] S. Chanda, A. Misha, S. Pal, Advancing language identification in code-mixed tulu texts: Harnessing deep learning techniques., in: FIRE (Working Notes), 2023, pp. 223–230.

[7] S. Chanda, S. Pal, Irlab@ iitbhu@ dravidian-codemix-fire2020: Sentiment analysis for dravidian languages in code-mixed text., in: FIRE (Working Notes), 2020, pp. 535–540.

[8] S. Chanda, A. Mishra, S. Pal, Sentiment analysis of code-mixed dravidian languages leveraging

pretrained model and word-level language tag, Natural Language Processing (2024) 1–23. doi:10.1017/nlp.2024.30.

[9] S. Chanda, R. Singh, S. Pal, Is meta embedding better than pre-trained word embedding to perform sentiment analysis for dravidian languages in code-mixed text?, Working Notes of FIRE (2021).

[10] S. Chanda, S. Sheth, S. Pal, Coarse and fine-grained conversational hate speech and offensive content identification in code-mixed languages using fine-tuned multilingual embedding, in: Forum for Information Retrieval Evaluation (Working Notes)(FIRE). CEUR-WS. org, 2022, pp. 502–512.

[11] S. Chanda, S. Pal, The effect of stopword removal on information retrieval for code-mixed data obtained via social media, SN Computer Science 4 (2023) 494.

[12] S. Amir, B. C. Wallace, H. Lyu, P. Carvalho, M. J. Silva, Modelling context with user embeddings for sarcasm detection in social media, in: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, Association for Computational Linguistics, Berlin, Germany, 2016, pp. 167–177. URL: https://aclanthology.org/K16-1017. doi:10.18653/v1/K16-1017.

[13] S. G. Wicana, T. Y. Ibisoglu, U. Yavanoglu, A review on sarcasm detection from machine-learning perspective, 2017 IEEE 11th International Conference on Semantic Computing (ICSC) (2017) 469–476. URL: https://api.semanticscholar.org/CorpusID:16074739.

[14] K. Jain, A. Deshpande, K. Shridhar, F. Laumann, A. Dash, Indic-transformers: An analysis of transformer language models for indian languages, 2020. arXiv:2011.02323.

[15] D. K. Jain, A. Kumar, G. Garg, Sarcasm detection in mash-up language using soft-attention based bi-directional lstm and feature-rich cnn, Appl. Soft Comput. 91 (2020) 106198. URL: https://api.semanticscholar.org/CorpusID:216439240.

[16] M. A. Galal, A. H. Yousef, H. H. Zayed, W. Medhat, Arabic sarcasm detection: An enhanced fine-tuned language model approach, Ain Shams Engineering Journal 15 (2024) 102736. URL: https://doi.org/10.1016/j.asej.2024.102736.

[17] N. N, V. V, A. Hegde, H. L. Shashirekha, Learning models with text augmentation for sarcasm detection in malayalam and tamil code-mixed texts, in: Forum for Information Retrieval Evaluation (Working Notes)(FIRE). CEUR-WS. org, 2023, pp. 287–298.

[18] M. Rahguouy, H. B. Giglou, T. Rahgooy, C. D. Seals, Null at semeval-2022 task 6: Intended sarcasm detection using stylistically fused contextualized representation and deep learning, in: Proceedings of the 16th International Workshop on Semantic Evaluation, 2022, pp. 862–870.

[19] A. Agrawal, A. An, M. Papagelis, Leveraging transitions of emotions for sarcasm detection, Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (2020). URL: https://api.semanticscholar.org/CorpusID:220729631.

[20] A. Pandey, D. K. Vishwakarma, Multimodal sarcasm detection (msd) in videos using deep learning models, in: 2023 International Conference in Advances in Power, Signal, and Information Technology (APSIT), 2023, pp. 811–814. doi:10.1109/APSIT58554.2023.10201731.

[21] 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, FIRE 2023 16 (2023) 22.

[22] 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, pp. 231–239.

[23] 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, pp. 231–239.

[24] 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.