

# Sarcasm Detection in Dravidian Code-Mixed Text Using Transformer-Based Models

Anik Basu Bhaumik<sup>1</sup>, Mithun Das<sup>2</sup>

<sup>1</sup>A.K Choudhury School of Information Technology, University of Calcutta, Kolkata, West Bengal, India

<sup>2</sup>Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, West Bengal, India

## Abstract

The words we express derive most of their meaning from the context we convey rather than their literal interpretation. Sarcasm represents one of the intriguing facets of human language, where the communicated message often carries a meaning different from the literal one, making it imperative to learn the underlying context. Failure to detect sarcasm can lead to misunderstandings, especially in text-based communication. In some cases, misinterpreting sarcasm as a genuine statement can lead to confusion or conflicts. Detecting sarcasm can help prevent such miscommunications. Further, users communicating on social media often use code-mixed texts in their posts. Transformer-based language models have demonstrated remarkable capabilities recently by harnessing their robust embedding representation and self-attention techniques, thereby expanding the horizons of language understanding. In this paper, we explored transformer-based machine learning models, namely mBERT and MURIL, for detecting sarcasm of code-mixed text in Dravidian languages (Tamil-English and Malayalam-English) at *Dravidian-CodeMix - FIRE 2023*. Our best-performing model MURIL, achieved the **first** position in the Tamil-English subtask (Macro-F1: 0.781) and secured the **second** position in the Malayalam-English subtask (Macro-F1: 0.731).

## Keywords

Code Mixed, Sarcasm Detection, Transformers, Natural Language Processing, Social Media

## 1. Introduction

The rapid proliferation of users in social media platforms has caused a transformative shift in the realm of human communication. These platforms provide individuals unprecedented opportunities to connect, share, and express themselves across diverse linguistic and cultural boundaries [1]. Sarcasm, a form of verbal irony wherein a person expresses something contrary to their true meaning, has become increasingly prevalent in this digital age[2]. It serves as a tool for mockery, ridicule, or the expression of contempt and scorn. In social media, sarcasm often relies on contextual cues, emojis, or specific textual markers to indicate that the message should not be taken literally. Instead, it is intended as a form of humor, critique, or irony. This mode of expression is pervasive in online interactions and serves a multitude of purposes, ranging from humor to commentary on a wide array of topics.

---


Forum for Information Retrieval Evaluation, December 15–18, 2023, Goa, India

✉ anikbb@gmail.com (A. B. Bhaumik); mithundas@iitkgp.ac.in (M. Das)

🆔 0000-0003-1442-312X (M. Das)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

| Text   | English meaning   | Label         |
|--|---|---------------|
| காடுவெட்டியார்<br>ஆசியூடன் படம் வெற்றி<br>பெற வாழ்த்துக்கள்  | Wish the film success with<br>the blessings of Kaduvettiar  | Non Sarcastic |
| Adapavi matha padathula<br>nadichinae itha epo da direct<br>pana.. Samuthirakani                               | When was the actress in<br>Adapaavi Madha's film<br>directed money..<br>Samudhirakani   | Non Sarcastic |
| Inimae ajith fans evanadhu....<br>Copy adikuradha pathi<br>pesuvenga... Anyway.. movie<br>hit aaga vazhthukkal | Regardless, wish<br>congratulations on the<br>movie's success, but please<br>refrain from discussing<br>copying among Ajith fans<br>from now on | Sarcastic     |

**Table 1**  
Example of Tamil-English texts with machine-generated English translation

Understanding sarcasm in social media holds the potential to offer valuable insights into public sentiment and emerging trends. Misinterpreting sarcasm as a sincere statement can lead to confusion or even conflicts in online exchanges. Detecting sarcasm, however, can be pivotal in averting such miscommunications. Sarcasm detection presents a unique challenge in the field of Natural Language Processing (NLP), particularly in comparison to straightforward statements. Unlike spoken language, written text lacks vocal cues and auditory inflections that often accompany sarcasm, making it even more challenging to discern. Additionally, variations in sarcasm's interpretation across diverse cultures and social groups, coupled with the dynamic evolution of language, contribute to the complexity of building comprehensive sarcasm detection models. Sarcasm detection can have valuable applications in social media platforms such as Twitter and Facebook, enabling a deeper understanding of an individual's perspective within a specific context.

Amidst this digital revolution, blending languages within social media discourse, often called "code-mixing", has emerged as a prominent linguistic phenomenon. Code-mixing occurs when individuals seamlessly incorporate multiple languages within a single communicative context, reflecting the rich tapestry of multicultural and multilingual societies. Detecting code-mixed sarcasm text presents unique challenges [3] in natural language processing (NLP) and sentiment analysis, owing to the intricate interplay between linguistic diversity, sarcasm, and cultural nuances.

To foster research and engagement in sarcasm detection, the organizers of the "Dravidian-CodeMix" [4, 5, 6] shared task at the FIRE 2023 conference have introduced a gold standard corpus for sarcasm and sentiment detection within code-mixed text in Dravidian languages (Tamil-English and Malayalam-English). The shared task aims to develop methodologies for the automatic detection of sarcastic text in Tamil-English and Malayalam-English languages. Table 1 & 2 provides some examples of such posts.

This paper explores existing transformer-based models, namely m-BERT [7] and MURIL [8], for our classification task. These models have consistently outperformed several baselines and

| Text  | English meaning  | Label         |
|---|--|---------------|
| അനിയത്തി പ്രാവീന് ശേഷം ഒരു സിനിമയും ചെയ്യല്ലെ എന്ന് പറഞ്ഞ ആള് ആണ് ഇപ്പോ എത്ര പടം ചെയ്തു | Praveen, who said he would never make a movie, has now made how many films?    | Non Sarcastic |
| ഇതൊപ്പൊ പാട്ട്..... Prrrrr<br>mammootty ude എറുവും വലിയ പടമല്ലെ ഇത് 🤔🤔🤔🤔🤔🤔              | "This isn't a song; it's not Mammootty's biggest film; it's just this!" 🤔🤔🤔🤔🤔🤔 | Sarcastic     |
| ഓ വല്ല സാദവം ഒന്നും അല്ല  | Oh, that's a good point, it's not happening at all.                            | Sarcastic     |

**Table 2**  
Example of Malayalam-English texts with machine-generated English translation

are regarded as state-of-the-art for various downstream tasks [9, 10]. Our approach involves pre-processing, hyper-parameter tuning, and other relevant techniques to construct our model. Among our models, MURIL has achieved **first** in the Tamil-English subtask and **second** in the Malayalam-English subtask.

## 2. Related Work

Numerous studies have been conducted in the field of sarcasm detection. Alita et al. [11] employed traditional models, including Random Forest Classifier, Naïve Bayes Classifier, and Support Vector Machine. On the other hand, Pandey et al. [12] introduced a novel approach—a hybrid attention-based Long Short Term Memory (HA-LSTM) network—tailored for identifying sarcastic statements. What sets this HA-LSTM network apart from conventional LSTM models is its incorporation of 16 distinct linguistic features within its hidden layers. Shrawankar and Chandankhede [13] delved into sarcasm detection within the context of workplace stress management. Their motivation stemmed from realizing that people often resort to sarcasm during verbal communication, through gestures, emoticons, or when writing reviews and comments. Such behaviors can escalate anxiety and even lead to depression. Majumder et al. [14] presented a multi-task learning-based framework utilizing a deep neural network for both sentiment and sarcasm Classification. Impressively, their method surpassed the state-of-the-art by 3-4% in benchmark datasets. Naz et al. [15] extracted data from Twitter to automatically identify sarcasm within customer reviews. Jain et al. [16] devised a hybrid model that combines bidirectional-LSTM, a softmax attention layer, and a CNN to identify sarcastic statements. To create their dataset, they collected 6,000 tweets from various domains, including government, politics, and entertainment. In 2021, Bedi et al. [17] proposed a multimodal system capable of detecting both sarcasm and humor in conversational dialogues. Their dataset featured a blend of Hindi and English conversations.

Bharti et al. [18] proposed a part-of-speech (POS) tagged approach, drawing data from Telugu

comedy TV shows. They categorized each data point into four patterns: “Normal Question followed by Normal Reply”, “Normal Question followed by Sarcastic Reply”, “Sarcastic Question followed by Sarcastic Reply”, and “Sarcastic Question followed by Normal Reply”. Based on these annotations, they selected 5,044 sarcastic statements and devised rules for sarcasm detection using POS tags. They also explored traditional machine learning techniques with different data splits, using annotated POS-tagged data as a feature set. Potamias et al. [19] introduced the RCNN-RoBERTa model, combining the rich embeddings of the RoBERTa network with one layer of RNN to capture inter-embedding dependencies and another layer of 1D convolution to address the bias in RNN-processed data. Their model achieved an impressive 91% accuracy with an F1 score of 0.90. Recently, transformer-based language models [20] such as BERT, m-BERT [7], and MURIL [8], have gained popularity in various downstream tasks, including classification and spam detection. These transformer models have consistently outperformed traditional deep-learning models like CNN-GRU and LSTM [21]. Recognizing the superior performance of transformer-based models, we have focused on building and deploying these models for our classification problem.

### 3. Dataset Description

The *Dravidian-CodeMix* shared task at **FIRE 2023** centers around identifying sarcastic comments/posts in code-mixed Dravidian languages gathered from social media platforms, primarily YouTube. This task poses a classification challenge, with the primary goal being the development of methodologies for detecting sarcasm in Tamil-English and Malayalam-English language contexts. Tamil, spoken by Tamil people in India and Sri Lanka and by the Tamil diaspora globally, holds official recognition in India, Sri Lanka, and Singapore. On the other hand, Malayalam is a Dravidian language predominantly spoken in the Indian state of Kerala. Each comment/post in the dataset comes with sentiment polarity annotations at the comment/post level. It is worth noting that this dataset mirrors real-world scenarios by exhibiting class imbalance issues. It encompasses all three types of code-mixed sentences: Inter-Sentential switch, Intra-Sentential switch, and Tag switching. The majority of comments are composed in both native script and Roman script, incorporating either Tamil or Malayalam grammar alongside an English lexicon or, conversely, English grammar paired with Tamil or Malayalam lexicon. Additionally, some comments are composed in the Tamil or Malayalam script, with intermittent English expressions.

#### 3.1. Tamil-English

Table 3 displays the class distribution within the Tamil-English language dataset, providing a clear breakdown of non-sarcastic and sarcastic categories. The dataset comprises a total of 42,244 instances, with 27,036 instances allocated for training. Among these, non-sarcastic comments account for 73.5%, while sarcastic comments make up the remaining 26.5%. Furthermore, there are 6,759 instances designated for validation and 8,449 for testing purposes. Evidently, the dataset exhibits a significant imbalance, mirroring real-world scenarios.

| Data              | Class         |           |        |
|-------------------|---------------|-----------|--------|
|                   | Non sarcastic | Sarcastic | Total  |
| <b>Train</b>      | 19,866        | 7,170     | 27,036 |
| <b>Validation</b> | 4,939         | 1,820     | 6,759  |
| <b>Test</b>       | 6,186         | 2,263     | 8,449  |
| <b>Total</b>      | 30,991        | 11,253    | 42,244 |

**Table 3**  
Dataset statistics for the Tamil-English language

### 3.2. Malayalam-English

Table 4 illustrates the class distribution in the Malayalam-English language dataset, offering a transparent delineation of non-sarcastic and sarcastic categories. The dataset encompasses a total of 18,840 instances, with 12,057 instances earmarked for training. Among these, non-sarcastic comments constitute 81.27%, while sarcastic comments constitute the remaining 18.73%. Additionally, there are 3,015 instances set aside for validation and 3,768 for testing purposes. It is evident that the dataset prominently displays a substantial imbalance, accurately reflecting real-world scenarios.

| Data              | Class         |           |        |
|-------------------|---------------|-----------|--------|
|                   | Non sarcastic | Sarcastic | Total  |
| <b>Train</b>      | 9,798         | 2,259     | 12,057 |
| <b>Validation</b> | 2,427         | 588       | 3,015  |
| <b>Test</b>       | 3,083         | 685       | 3,768  |
| <b>Total</b>      | 15,308        | 3,532     | 18,840 |

**Table 4**  
Dataset statistics for the Malayalam-English language

## 4. System Description

This section discusses the preprocessing steps and various models that we implement for the task of sarcasm detection.

### 4.1. Problem formulation

We formulate the sarcasm detection task in this paper as follows. Given a dataset  $\mathbf{D}$  comprising pairs  $(\mathbf{X}, \mathbf{Y})$ , where  $\mathbf{X} = w_1, w_2, \dots, w_m$  represents a text sample consisting of a sequence of words, and  $\mathbf{Y}$  denotes its corresponding label, the primary objective is to train a classifier  $F : F(\mathbf{X}) \rightarrow \mathbf{Y}$ . This classifier should be capable of accurately determining the presence or absence of sarcasm in previously unseen text samples, where  $\mathbf{Y} \in \{0, 1\}$  serves as the ground-truth label. Here, 0 signifies non-sarcastic, while 1 indicates sarcastic.

## 4.2. Preprocessing

Before developing the models, we embark on a series of preprocessing steps to ready the data for sarcasm detection. We employ a blend of custom functions and useful libraries, including “emoji” and “nltk”, to carry out essential preprocessing tasks. The ensuing preprocessing steps are as follows –

- *Replacing Tagged User Names:* We substitute all tagged user names with the “@user” token to eliminate personal identifiers from the text.
- *Removing Non-Alphanumeric Characters:* We remove non-alphanumeric characters from the text, with the exception of full stops and certain punctuation marks like ‘|’, and ‘.’. This step aims to ensure that the machine can identify the sequence of characters accurately.
- *Convert Emojis, Flags, and Emotions:* We also convert emojis, flags, and emotions in the text into their textual representations.
- *Removing URLs:* All URLs are excised from the text to exclude any web links that may not be relevant to sarcasm detection.
- *Keeping Hashtags:* We preserve hashtags within the text since they often contain contextual information that can prove valuable for identifying sarcasm.

By executing these preprocessing steps, we ensure that the text data is clean and optimized for the subsequent classification task.

## 4.3. Models

**MURIL:** MURIL[8] or Multilingual Representations for Indian Languages <sup>1</sup>, is a specialized language model that is a transformer encoder consisting of 12 layers with 12 attention heads. The model has been trained on 17 Indian languages and their transliterated counterparts using the MLM (Masked Language Model) and the NSP (Next Sentence Prediction) loss functions. The monolingual documents while training MuRIL was trained using MLM while the translated and transliterated pairs were trained using Translation language modeling (TLM). The dataset utilized for pre-training was acquired from publicly available corpora sourced from Wikipedia, Common Crawl, PMINDIA, and Dakshina. This model was developed as a solution to several gaps in other multilingual LMs, like smaller representations of Indian languages also it was aimed to capture several nuances of the Indian language through its code-mixed transliterated data points for training.

**m-BERT:** m-BERT[7] or Multilingual BERT <sup>2</sup>, is a state-of-the-art language model developed by Google. It utilizes the BERT (Bidirectional Encoder Representations from Transformers) architecture and is trained on a massive multilingual corpus (104 languages), enabling it to comprehend and process text in numerous languages. m-BERT’s architecture allows for seamless cross-lingual transfer of knowledge and context, making it a versatile tool for several natural language processing tasks across various languages. Its pre-trained representations have significantly progressed multilingual NLP research and applications.

---

<sup>1</sup><https://huggingface.co/google/muril-base-cased>

<sup>2</sup><https://huggingface.co/bert-base-multilingual-uncased>

| Model  | Acc          | M-F1         | F1(S)        | P(S)         | R(S)         | ROC-AUC      |
|--------|--------------|--------------|--------------|--------------|--------------|--------------|
| MURIL  | <b>0.781</b> | <b>0.743</b> | <b>0.644</b> | <b>0.571</b> | <b>0.738</b> | <b>0.767</b> |
| m-BERT | 0.776        | 0.737        | 0.637        | 0.563        | 0.733        | 0.762        |

**Table 5**

Performance comparisons of each model for Tamil-English language. P: Precision. R: Recall. S: Sarcastic. M: Macro. The best performance between both models is indicated in **bold** for each column.

| Model  | Acc          | M-F1         | F1(S)        | P(S)         | R(S)         | ROC-AUC      |
|--------|--------------|--------------|--------------|--------------|--------------|--------------|
| MURIL  | <b>0.850</b> | <b>0.731</b> | <b>0.553</b> | <b>0.604</b> | 0.510        | 0.718        |
| m-BERT | 0.813        | 0.709        | 0.536        | 0.489        | <b>0.594</b> | <b>0.728</b> |

**Table 6**

Performance comparisons of each model for Malayalam-English language. P: Precision. R: Recall. S: Sarcastic. M: Macro. The best performance between both models is indicated in **bold** for each column.

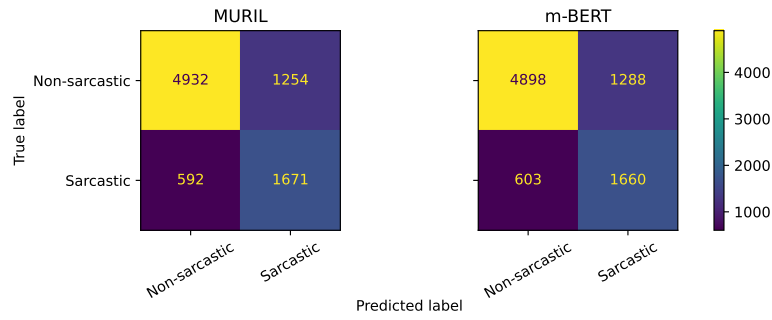
#### 4.4. Tuning Parameters

We employed the same set of hyperparameters for our transformer-based models for both languages. We conducted training over ten epochs using the Adam optimizer[22] and binary cross entropy loss function, initializing with a learning rate of 2e-5 and setting adam\_epsilon to 1e-8. Fine-tuning was accomplished with a batch size of 16, and we constrained the number of tokens processed to the model to 128. Our model checkpoint selection was based on the highest validation performance, specifically in terms of the macro F1 score. Using these saved checkpoints, we made predictions on the test set. These hyperparameters were chosen following existing literature on similar tasks [23, 24, 25]. All model implementations were carried out in Python, utilizing the PyTorch library.

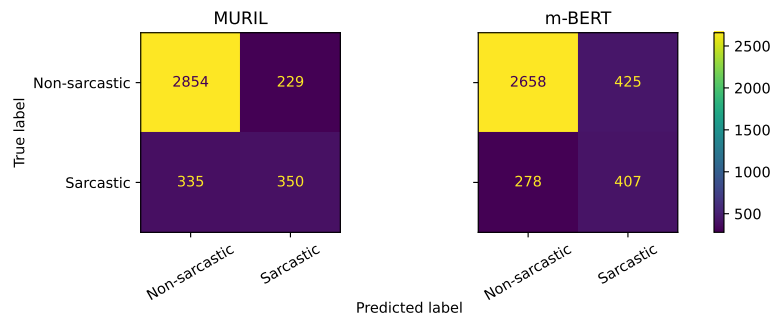
## 5. Results

In Tables 5 and 6, we present the performance for both models across both languages. We observed that, for the Tamil-English language, the MURIL(Acc: 0.781, M-F1: 0.743) model outperforms the m-BERT(Acc: 0.776, M-F1: 0.737) model across all metrics. For the Malayalam-English language, It was noted that although the m-BERT model achieves higher scores for the ROC-AUC(m-BERT: 0.728, MURIL: 0.718) and Recall(m-BERT:0.594, MURIL: 0.510) metric in the Sarcastic class, MURIL(Acc:0.850, M-F1: 0.731) outperforms m-BERT(Acc:0.813, M-F1: 0.709) across all other metrics. Despite m-BERT being pre-trained on a larger dataset, MURIL’s enhanced performance can be attributed to its specific pre-training on Indian languages and their transliterated counterparts. This specialization empowers MURIL with an improved ability to comprehend code-mixed texts in Indic languages compared to m-BERT. The confusion matrix of each model is shown for both languages in Figure 1 and 2.





**Figure 1:** Confusion-matrix for Tamil-English



**Figure 2:** Confusion-matrix for Malayalam-English

## 6. Conclusions

In this shared task, we tackle the novel challenge of identifying sarcastic comments/posts in code-mixed Dravidian languages, specifically Tamil-English and Malayalam-English, collected from social media. To evaluate performance, we leveraged transformer-based models such as m-BERT and MURIL. Our findings demonstrated that MURIL outperforms m-BERT across several metrics in both languages. MURIL’s superior performance can be attributed to its specialized pre-training in Indian languages and their transliterated counterparts. Our team, “**hate-alert**”, secured the **first** position in the Tamil-English subtask and achieved **second** in the Malayalam-English subtask. Our performing model, MURIL, attained a Macro-F1 score of 0.743 for Tamil-English and a Macro-F1 score of 0.731 for Malayalam-English. In the future, we intend to explore additional transformer-based models and recent Large Language Models (LLM) to enhance our approach in this domain further.

## References

- [1] M. Das, B. Mathew, P. Saha, P. Goyal, A. Mukherjee, Hate speech in online social media, ACM SIGWEB Newsletter 2020 (2020) 1–8.
- [2] R. Pandey, J. P. Singh, Bert-lstm model for sarcasm detection in code-mixed social media post, Journal of Intelligent Information Systems 60 (2023) 235–254.



- [3] G. Chittaranjan, Y. Vyas, K. Bali, M. Choudhury, Word-level language identification using crf: Code-switching shared task report of msr india system, in: Proceedings of The First Workshop on Computational Approaches to Code Switching, 2014, pp. 73–79.
- [4] B. R. Chakravarthi, Hope speech detection in youtube comments, *Social Network Analysis and Mining* 12 (2022) 75.
- [5] B. R. Chakravarthi, A. Hande, R. Ponnusamy, P. K. Kumaresan, R. Priyadarshini, How can we detect homophobia and transphobia? experiments in a multilingual code-mixed setting for social media governance, *International Journal of Information Management Data Insights* 2 (2022) 100119.
- [6] 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.
- [7] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4171–4186.
- [8] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. K. Margam, P. Aggarwal, R. T. Nagipogu, S. Dave, et al., Muril: Multilingual representations for indian languages, *arXiv preprint arXiv:2103.10730* (2021).
- [9] M. Das, S. Banerjee, P. Saha, A. Mukherjee, Hate speech and offensive language detection in bengali, in: Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing, 2022, pp. 286–296.
- [10] M. Das, P. Saha, B. Mathew, A. Mukherjee, Hatecheckin: Evaluating hindi hate speech detection models, in: Proceedings of the Thirteenth Language Resources and Evaluation Conference, 2022, pp. 5378–5387.
- [11] D. Alita, S. Priyanta, N. Rokhman, Analysis of emoticon and sarcasm effect on sentiment analysis of indonesian language on twitter, *Journal of Information Systems Engineering and Business Intelligence* 5 (2019) 100–109.
- [12] R. Pandey, A. Kumar, J. P. Singh, S. Tripathi, Hybrid attention-based long short-term memory network for sarcasm identification, *Applied Soft Computing* 106 (2021) 107348.
- [13] U. Shrawankar, C. Chandankhede, Sarcasm detection for workplace stress management, *International Journal of Synthetic Emotions (IJSE)* 10 (2019) 1–17.
- [14] N. Majumder, S. Poria, H. Peng, N. Chhaya, E. Cambria, A. Gelbukh, Sentiment and sarcasm classification with multitask learning, *IEEE Intelligent Systems* 34 (2019) 38–43.
- [15] F. Naz, M. Kamran, W. Mehmood, W. Khan, M. S. Alkathairi, A. S. Alghamdi, A. A. Alshdadi, Automatic identification of sarcasm in tweets and customer reviews, *Journal of Intelligent & Fuzzy Systems* 37 (2019) 6815–6828.
- [16] D. Jain, A. Kumar, G. Garg, Sarcasm detection in mash-up language using soft-attention based bi-directional lstm and feature-rich cnn, *Applied Soft Computing* 91 (2020) 106198.
- [17] M. Bedi, S. Kumar, M. S. Akhtar, T. Chakraborty, Multi-modal sarcasm detection and humor

classification in code-mixed conversations, *IEEE Transactions on Affective Computing* (2021).

- [18] S. K. Bharti, R. Naidu, K. S. Babu, Hyperbolic feature-based sarcasm detection in telugu conversation sentences, *Journal of Intelligent Systems* 30 (2020) 73–89.
- [19] R. A. Potamias, G. Siolas, A.-G. Stafylopatis, A transformer-based approach to irony and sarcasm detection, *Neural Computing and Applications* 32 (2020) 17309–17320.
- [20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [21] B. Mathew, P. Saha, S. M. Yimam, C. Biemann, P. Goyal, A. Mukherjee, Hatexplain: A benchmark dataset for explainable hate speech detection, 2020. [arXiv:2012.10289](https://arxiv.org/abs/2012.10289).
- [22] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2019. [arXiv:1711.05101](https://arxiv.org/abs/1711.05101).
- [23] S. Banerjee, M. Sarkar, N. Agrawal, P. Saha, M. Das, Exploring transformer based models to identify hate speech and offensive content in english and indo-aryan languages, *arXiv preprint arXiv:2111.13974* (2021).
- [24] M. Das, S. Banerjee, P. Saha, Abusive and threatening language detection in urdu using boosting based and bert based models: A comparative approach, *arXiv preprint arXiv:2111.14830* (2021).
- [25] M. Das, S. Banerjee, A. Mukherjee, Data bootstrapping approaches to improve low resource abusive language detection for indic languages, in: *Proceedings of the 33rd ACM Conference on Hypertext and Social Media*, 2022, pp. 32–42.