Sarcasm Identification in Codemix Dravidian Languages

Prabhu Ram. N[*,†], Meera Devi. T[†], Kanisha. V[*,†], Meharnath. S[†] and Manoji. B[†]

Department of Electronics and Communication Engineering, Kongu Engineering College, Erode, TamilNadu

Abstract

Social media enables forms of communication using text, audio and video. The sarcastic nature of the text has an impact on an individual's well-being. It is crucial to determine whether the content is sarcastic or not. Social media posts often contain a mix of languages and address issues related to real-life situations. However, identifying sarcasm in the languages spoken in India, which has 22 languages, can be more challenging due to extensive borrowing of vocabulary. The dataset used for this study includes code-mixed languages such as Tamil-English and Malayalam-English. The training, validation, and test datasets are proportionally divided with labels to indicate whether the text is sarcastic. We performed experiments using the transfer learning model and observed that the BERT model gave the best result.

Keywords

BERT, Codemix, Dravidian Language, NLP, Sarcasm, Transformer

1. Introduction

In the 21st century, social media has become a platform with over 4.9 billion users [1]. It enables people to easily share their thoughts and opinions, making communication faster and more convenient. However, the content on media can have impacts such as anxiety, depression and even suicidal thoughts among individuals. Unfortunately, some individuals use it to criticize or insult others, employing sarcasm that can be difficult to detect because of its implicitness. People now frequently use irony due to the increasing availability of internet connections and numerous new applications. Consequently, many countries have implemented social media surveillance measures to monitor citizens [2]. However, social media posts often involve a mix of languages, while Dravidian languages like Tamil and Malayalam add complexity due to their nuances. People utilize Natural Language Processing (NLP), a branch of artificial intelligence (AI) that analyses human language patterns, to identify sarcasm [3]. This paper explores the challenges and opportunities in sarcasm identification in codemixed Dravidian languages. In the subsequent sections, we will delve into the historical context of methods used in solving related problems, the methodology involving data pre-processing and modelling, the experimental

FIRE 2023: 15th meeting of the Forum for Information Retrieval Evaluation, December 15 – 18, 2023, India

[*] Corresponding author.

[†] These authors contributed equally.

prabhuramnphd@gmail.com (P. Ram. N); tmeeradevi@gmail.com (M. Devi. T); kanishav.20ece@kongu.edu (Kanisha. V); meharnaths.20ece@kongu.edu (Meharnath. S); manojib.20ece@kongu.edu (Manoji. B)

0000-0003-2769-9790 (P. Ram. N); 0000-0003-4989-4028 (M. Devi. T)

setup for fine-tuning the pre-trained model, and the statistical analysis of our results with discussion [24].

2. Related work

Natural language processing researchers use various techniques to tackle the challenging task of sarcasm detection. Although hate speech and sarcasm exhibit indirect connections, the pre-trained multilingual BERT model's contextual understanding shows a more significant similarity due to its training on multilingual Wikipedia language sources [4, 5]. Hate speech detection for English, German and Hindi Languages using the Multilingual BERT model uses Machine Learning (ML) and Deep Learning approaches (DL). We employed various classification architectures based on networks, including models like subword level LSTM, Hierarchical LSTM, BERT, XLM-RoBERTa, LSTM, GRU, and XLNet [6, 7, 8]. Additionally, we utilized machine learning-based classification models such as Support Vector Machine (SVM) Logistic Regression (LR) Random Forest Classifier (RFC) [8, 9, 10, 11, 12] and K-Nearest Neighbour (KNN) [13]. Among these models used for the code mix, the Tamil dataset classification task SVM model showed performance compared to machine learning models. We have also employed RNN and MLP deep learning models to enhance classification. TFIDF (Term-Frequency-Inverse-Document-Frequency) [14] serves as the text preprocessor, and SVM (Support Vector Machines) [15] functions as the classifier. A two-phase approach for sarcasm detection using machine learning algorithms involves feature extraction, feature selection, and classification using support vector machines [16]. Sarcasm detection using a hierarchical attention network that incorporates both word and sentence-level attention mechanisms, using a graph convolutional network that includes both syntactic and semantic information, using a bootstrapping approach that iteratively learns new sarcastic patterns from labelled data achieved state-of-the-art performance on several benchmark datasets. In the context of emotion recognition, a study [7] focused on enhancing the effectiveness of BERT word embeddings through knowledge-based fine-tuning techniques. This research underscores the ongoing sentiment analysis and emotion recognition efforts within natural language processing. The model is connected with the fully connected network with a softmax activation function which is used to classify the emotions of the given sentence. A hybrid model of bidirectional LSTM with a softmax attention layer and convolution neural network for real-time sarcasm detection in code-switched tweets achieved a superior classification accuracy of 92.71% and F1-measure of 89.05% [17]. The model trained on larger datasets achieved higher efficiency, and the inclusion of Dravidian Languages in the dataset will be helpful for this task [7, 13, 18].

3. Methodology

The step-by-step process for conducting the experiment has been thoroughly explained and outlined in the Figure ??. These sections provide a comprehensive description of how the experiment was carried out, ensuring a clear and detailed understanding of the experimental flow. The process commences with the selection and initialization of a pre-trained model. Subsequently, the dataset is collected and pre-processed. Rigorous evaluation is performed

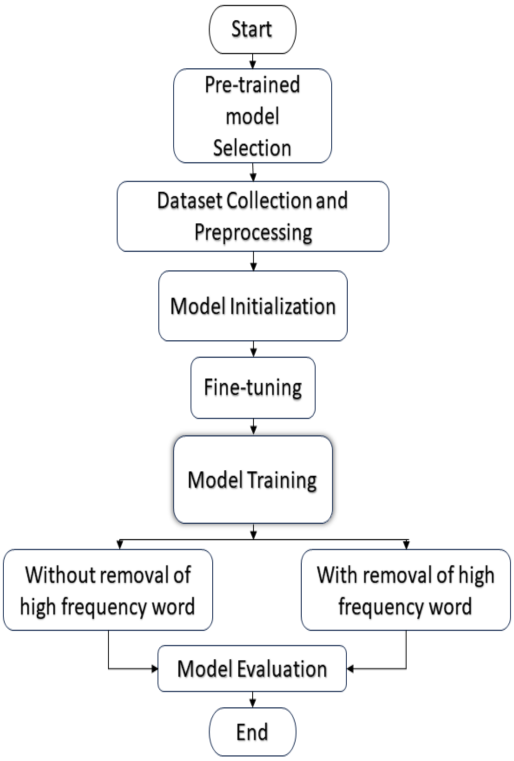
Start
Pre-trained model Selection
Dataset Collection and Preprocessing
Model Initialization
Fine-tuning
Model Training
Without removal of high frequency word
With removal of high frequency word
Model Evaluation
End

Figure 1: Block diagram.

on a designated validation and testing dataset, and fine-tuning strategies, including transfer learning, are employed to tailor the model for the specific task. Furthermore, hyperparameter optimization techniques are applied to enhance the model's performance. The ensuing section delves into a comprehensive presentation of results and constructive discussions, highlighting the efficacy of fine-tuning and the implications of hyperparameter adjustments.

3.1. Dataset collection

The datasets for Tamil-English and Malayalam-English is included in the work, which is the collection of comments from YouTube video. Data on all three types of code interlinked sentences: Inter-Sentential switch, Intra-Sentential switch and Tag Swapping is included in the dataset. Most comments were written in native script and Roman script with either Tamil / Malayalam grammar with English lexicon or English grammar with Tamil / Malayalam lexicon. Some of the comments have been written in Tamil and Malayalam, with English translations between them.

Table 1
Dataset Distribution

Label	Training dataset count	
	English-Tamil dataset	English-Malayalam dataset
Sarcastic	7170	2259
Non-Sarcastic	19866	9798
Total length	27036	12057

3.2. Data preprocessing

The dataset consists of text samples categorized as either 'Sarcastic' or 'Non-Sarcastic' and is encoded into values assigning 0 and 1 for 'Sarcastic' and 'Non-Sarcastic' labels, respectively. The maximum word length of the dataset is fixed to 128 tokens of words [19, 20].

3.3. Model training

For training the model, Simple Transformers is employed. Utilization of Bert-base-uncased model architecture is done because of its ability to capture information from the text data effectively, and this architecture is best suited for Binary Classification. The model is trained to classify the text either as 'Sarcastic' or 'Non-Sarcastic'.

3.4. Model evaluation and training

The performance of the trained model is validated and assessed using the test dataset. Based on the predictions made by the model, a classification report is generated that includes accuracy, f1 score, precision and recall. The model may or may not be biased towards a particular class because of class imbalances, as shown in Table 1. So, the highly frequent tokens from the utterance of the Sarcastic class with respect to the Non-Sarcastic class are taken from the training dataset. The high-frequent tokens are removed from the test dataset, and a classification report is generated to check whether the model is influenced by the high-frequent tokens.

4. Experimental setup

Three datasets are utilized for the task of sarcasm identification for each English-Tamil and English-Malayalam dataset.

The training dataset consists of a larger number of 'Non-Sarcastic' labels than the 'Sarcastic' labels addressing class imbalance, as shown in Table 1. This huge difference in the occurrence of the labels makes the model more biased towards the 'Non-Sarcastic' texts.

5. Results and discussion

Precision, recall, and F1-score scores for the bert-based-uncased sarcasm detection model were competitive. The performance of the majority class (Non-sarcastic) and the minority class (Sarcastic), however, showed a clear discrepancy. The bert-base-uncased model tends to favour

Table 2
Classification Report before removing the high frequent tokens

	English-Tamil dataset			English-Malayalam dataset		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Sarcastic class	0.63	0.49	0.55	0.46	0.10	0.17
Non-sarcastic class	0.83	0.89	0.86	0.83	0.97	0.90
Accuracy			0.79			0.81
Macro average	0.73	0.69	0.70	0.64	0.54	0.53
Weighted average	0.77	0.79	0.78	0.76	0.81	0.76

the majority class, like many other machine learning algorithms. As a result, the minority class had great accuracy but low recall (Sarcastic). A high F1 score indicates that the model can accurately detect sarcasm while minimising false positives (non-sarcastic text misclassified as sarcastic) and false negatives (sarcastic text misclassified as non-sarcastic). Tamil-English and Malayalam-English datasets shown in Table 2 have respective F1 values of 0.79 and 0.81. The removal of highly frequent words from the Sarcastic class with respect to the Non-Sarcastic class did not influence the model, as shown in Table 3. The differences in accuracy between the English-Tamil and English-Malayalam datasets before and after high-frequency token removal provide important information on how token removal affects model performance. The accuracy in the English-Tamil dataset is 0.79, meaning that 79% of instances are properly classified by the model before high-frequency tokens are eliminated. But it's clear that the accuracy has slightly decreased to 0.78 after the high-frequency tokens were eliminated. The possible cause of this little decrease in accuracy is the loss of important data that high-frequency tokens were carrying. It implies that these tokens may indeed contribute positively to the model's capacity to accurately identify instances in the English-Tamil dataset.

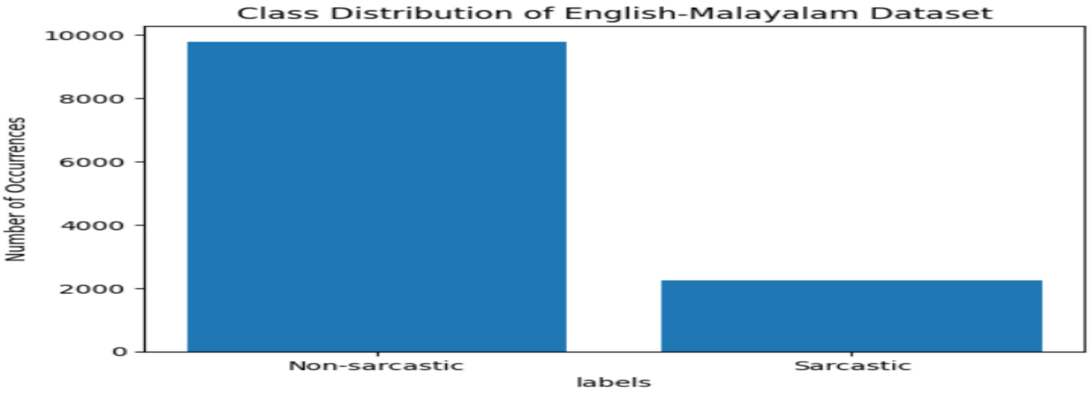

Figure 2: Confusion matrix on the given test dataset for English-Tamil dataset

On the other hand, prior to the removal of high-frequency tokens, the English-Malayalam dataset shows an accuracy of 0.81, showing a high degree of classification accuracy. Upon eliminating high-frequency tokens, the accuracy noticeably increases to 0.82. This improved accuracy raises the possibility that high-frequency tokens might contaminate the data with

	English-Tamil dataset				English-Malayalam dataset			
	Precision	Recall	F1-score	Support	Precision	Recall	F1-score	Support
Sarcastic	0.63	0.48	0.54	2263	0.59	0.04	0.07	685
Non-Sarcastic	0.82	0.90	0.86	6186	0.82	0.99	0.90	3083
Accuracy			0.78	8449			0.82	3768
Macro average	0.73	0.69	0.70	8449	0.71	0.52	0.49	3768
Weighted average	0.77	0.78	0.77	8449	0.78	0.82	0.75	3768

noise or ambiguity. Eliminating these tokens improves model accuracy by streamlining the dataset.

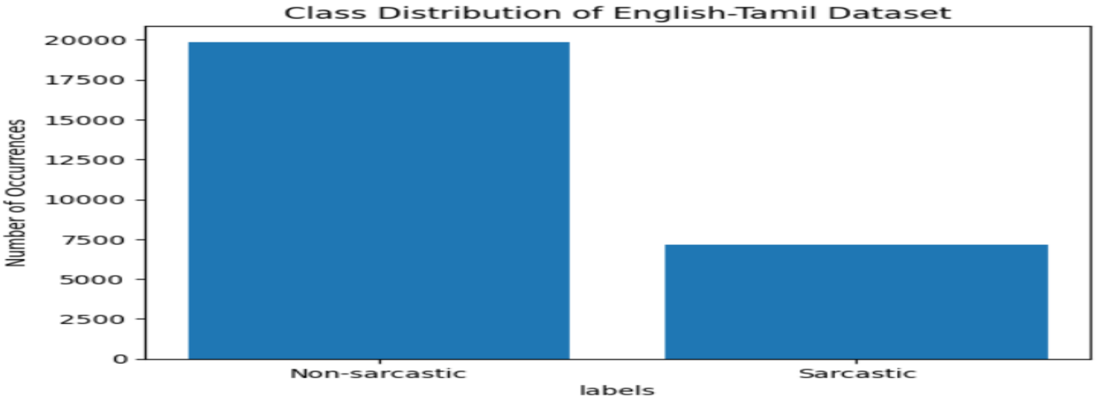

Figure 3: Confusion matrix on the given test dataset for English-Malayalam dataset

The differences in accuracy between the two datasets demonstrate how high-frequency token influence on model performance varies depending on the context. Their elimination improved the accuracy of the English-Malayalam dataset but somewhat decreased the accuracy of the English-Tamil dataset. These results highlight the need for a sophisticated strategy in NLP tasks when determining whether high-frequency tokens to keep or discard. Making educated judgments requires a thorough analysis of the linguistic properties of the dataset and the possibility of noise introduction from high-frequency tokens. Enhancing model accuracy in code-mixed Dravidian languages requires striking the ideal balance between information retention and data refining, and this approach should be dictated by the unique linguistic characteristics of each dataset. A confusion matrix is used to analyse the performance of the model, as shown in Figure 3. The task involves binary classification involving the 'Non-Sarcastic' and 'Sarcastic' classes. This matrix comprises four distinct categories: True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) [21]. Figure 3 and Figure 3 reveals the model's strengths and areas of improvement in the binary classification task, offering valuable insights for refining its performance and guiding future research efforts.

6. Conclusion and future work

Sarcasm identification using the BERT model provides a major improvement in natural language processing. Intricate language nuances can be captured by BERT, enabling it to capture contextual awareness and pre-trained knowledge, which makes it better for the complex task of sarcasm detection. The accuracy of the model is substantially improved by its capacity to take into account the larger context of a statement and comprehend how specific words or phrases fit into that context. BERT can provide more balanced predictions, reducing the tendency to misclassify sarcastic statements as non-sarcastic due to the imbalance issue. However, it's essential to acknowledge that class imbalance remains a challenge, and further research should focus on strategies like oversampling, undersampling, or using different loss functions to fine-tune BERT models effectively. By addressing this class imbalance problem, we can continue to enhance the accuracy and reliability of sarcasm detection using BERT-based models, making them more valuable in real-world applications. Due to the class imbalance problem, the model is more biased towards the Non-Sarcastic texts, which reduces the model's accuracy. In addressing the class imbalance, the potential integration of the Synthetic Minority Over-sampling Technique (SMOTE) algorithm holds significant promise [22]. By integrating SMOTE into the BERT model, we can create samples for the class. This helps in creating a balanced dataset during training and greatly improves the accuracy of the model.

References

[1] J. Belle Wong, Top social media statistics and trends of 2023, Online, 2023. URL: https://www.forbes.com/advisor/business/social-media-statistics/.

[2] B. E. Duffy, N. K. Chan, "you never really know who's looking": Imagined surveillance across social media platforms, New Media & Society 21 (2019) 119–138.

[3] B. Harish, R. K. Rangan, A comprehensive survey on indian regional language processing, SN Applied Sciences 2 (2020) 1204.

[4] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[5] 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.

[6] T. Santosh, K. Aravind, Hate speech detection in hindi-english code-mixed social media text, in: Proceedings of the ACM India joint international conference on data science and management of data, 2019, pp. 310–313.

[7] S. Banerjee, A. Jayapal, S. Thavareesan, Nuig-shubhanker@ dravidian-codemix-fire2020: Sentiment analysis of code-mixed dravidian text using xlnet, arXiv preprint arXiv:2010.07773 (2020).

[8] B. R. Chakravarthi, R. Priyadharshini, N. Jose, T. Mandl, P. K. Kumaresan, R. Ponnusamy, R. Hariharan, J. P. McCrae, E. Sherly, et al., Findings of the shared task on offensive language identification in tamil, malayalam, and kannada, in: Proceedings of the first workshop on speech and language technologies for Dravidian languages, 2021, pp. 133–145.

[9] A. Muneer, S. M. Fati, A comparative analysis of machine learning techniques for cyberbullying detection on twitter, Future Internet 12 (2020) 187.

[10] V. Pathak, M. Joshi, P. Joshi, M. Mundada, T. Joshi, Kbcnmujal@ hasoc-dravidian-codemix-fire2020: Using machine learning for detection of hate speech and offensive code-mixed social media text, arXiv preprint arXiv:2102.09866 (2021).

[11] B. R. Chakravarthi, N. Jose, S. Suryawanshi, E. Sherly, J. P. McCrae, A sentiment analysis dataset for code-mixed malayalam-english, arXiv preprint arXiv:2006.00210 (2020).

[12] P. R. Nagarajan, V. Mammen, V. Mekala, M. Megalai, A fast and energy efficient path planning algorithm for offline navigation using svm classifier, Int. J. Sci. Technol. Res 9 (2020) 2082–2086.

[13] B. R. Chakravarthi, V. Muralidaran, R. Priyadharshini, J. P. McCrae, Corpus creation for sentiment analysis in code-mixed tamil-english text, arXiv preprint arXiv:2006.00206 (2020).

[14] B. Subba, P. Gupta, A tfidfvectorizer and singular value decomposition based host intrusion detection system framework for detecting anomalous system processes, Computers & Security 100 (2021) 102084.

[15] V. Cherkassky, Y. Ma, Practical selection of svm parameters and noise estimation for svm regression, Neural networks 17 (2004) 113–126.

[16] S. M. Sarsam, H. Al-Samarraie, A. I. Alzahrani, B. Wright, Sarcasm detection using machine learning algorithms in twitter: A systematic review, International Journal of Market Research 62 (2020) 578–598.

[17] 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.

[18] P. K. Roy, S. Bhawal, C. N. Subalalitha, Hate speech and offensive language detection in dravidian languages using deep ensemble framework, Computer Speech & Language 75 (2022) 101386.

[19] B. R. Chakravarthi, Hope speech detection in youtube comments, Social Network Analysis and Mining 12 (2022) 75.

[20] 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, International Journal of Information Management Data Insights 2 (2022) 100119.

[21] O. Caelen, A bayesian interpretation of the confusion matrix, Annals of Mathematics and Artificial Intelligence 81 (2017) 429–450.

[22] B. R. Chakravarthi, D. Chinnappa, R. Priyadharshini, A. K. Madasamy, S. Sivanesan, S. C. Navaneethakrishnan, S. Thavareesan, D. Vadivel, R. Ponnusamy, P. K. Kumaresan, Developing successful shared tasks on offensive language identification for dravidian languages, arXiv preprint arXiv:2111.03375 (2021).