Learning Models with Text Augmentation for Sarcasm Detection in Malayalam and Tamil Code-mixed Texts

Navya N, Vanitha V, Asha Hegde and H L Shashirekha

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

Abstract

Users can express their sentiments, sarcasm, emotions, signs of depression, and hatred comments on social media platforms leading to user-generated text. Sarcasm is a form of linguistic expression that conveys a message which is opposite to the intended message, that is been typically used to mock or humorously criticize. If the sarcastic messages crosses the boundary, they can hurt an individual/community spoiling the healthy environment of social media platforms. The social media text usually will be in code-mixed format and sarcastic comments which are not excluded from this contributes to the complexities of processing code-mixed sarcastic comments. Further, in parallel with user-generated content on social media platforms, sarcastic comments also have increased making it difficult to detect them manually. Hence, there is a demand for tools/models that could automatically identify such code-mixed sarcastic comments on social media to keep the social media platforms healthy. In this paper, we - team MUCS, describe three distinct binary classification models: i) Ensemble of Machine Learning (ML) classifiers (Logistic Regression (LR), Random Forest Classifier (RF), Support Vector Classifier (SVC))) with hard voting, Deep Learning (DL) models (Convolutional Neural Network (CNN)), and Transfer Learning (TL) based models (Mutlilingual Bidirectional Encoder Representations from Transformers (mBert) and Distilled version of Mutlilingual Bert (mDistilBert) for Malayalam and Tamil code-mixed texts respectively), submitted to the shared task "Sarcasm Identification of Dravidian Languages (Malayalam and Tamil)" at DravidianCodeMix 2023 @Forum for Information Retrieval Evaluation (FIRE) 2023. As the given dataset is imbalanced, Text Augmentation (TA) techniques are explored to balance the dataset. Among the proposed models, Ensemble model obtained macro F1 scores of 0.71 and 0.70 securing 4th and 5th ranks for Malayalam and Tamil code-mixed texts respectively.

Keywords

Code-mixed, Machine Learning, Deep Learning, Transfer Learning, Text Augmentation

1. Introduction

In the internet era, social media platforms play a significant role in facilitating the sharing of user thoughts, reviews, and opinions. However, the anonymity on these platforms often leads to the presence of hate speech, offensive language, and sarcasm, in user-generated text targeting an individual or a group. Sarcasm, in particular, is a common way to mock or ridicule something or someone in which the face value of the text is just opposite to its intended meaning. For e.g., a comment like "Very good; well done!" for a bad situation or when something has really went wrong, depicts a sarcastic comment. The face value of this comment says someone has done a good job, however, the subtle meaning of this comment speaks exactly opposite to the face

Forum for Information Retrieval Evaluation, December 15-18, 2023, India

navyabangera451@gmail.com (N. N); vjvanitha001@gmail.com (V. V); hegdekasha@gmail.com (A. Hegde); hlsrekha@mangaloreuniversity.ac.in (H. L. Shashirekha)

value of the comment. As sarcastic comments may hurt the individuals/communities and spoil the healthy social media environment, detecting sarcastic content on social media has to be given at most priority. Sarcasm detection is the essential process of recognizing and flagging instances of sarcasm in social media texts, helping to moderate the content to have a respectful and safe online environment.

Social media text is often a collection of very informal user-generated text exhibiting code-mixing, especially in multilingual countries like India, where people commonly blend their mother tongue or local language with English when posting comments on social media platforms [1]. Code-mixing, as a linguistic phenomenon, entails the incorporation of multiple languages within a single sentence, word, or even at a sub-word level [2, 3]. This practice often stems from technological constraints that favor the use of the Roman script, making it easier for users to express their sentiments and opinions in their native languages while incorporating English terms. The convenience of keying in Roman letters is evident, as it avoids the complexities associated with using native language scripts, especially in the context of Indian languages that exhibit complex key combinations [4]. Further, the informal nature of social media text allows incomplete sentences or words, user-defined abbreviations, words with recurring characters, slangs, suffixes from other languages, etc., which varies from one user to another. This variation in user-generated text makes it very challenging to process them.

Tamil is a prominent Dravidian language spoken by Tamil people in India, Sri Lanka, and worldwide by the Tamil diaspora [5]. It has official recognition in India, Sri Lanka, and Singapore. Tamil script which evolved from the Tamili script, Vatteluttu alphabet, and Chola-Pallava script [6] comprises of 12 vowels, 18 consonants, and 1 āytam (voiceless velar fricative). It is also used for writing the minority languages - Saurashtra, Badaga, Irula, and Paniya. Malayalam, on the other hand, is a Dravidian language spoken mainly in Kerala, India. It uses an alpha-syllabic script that is part of the abugida family of writing systems, combining alphabetic and syllable-based elements [7]. Tamil/Malayalam speaking people, especially the younger generation who are active on social media use a combination of Tamil/Malayalam and English words to posts the comments using a combination of native and Roman script leading to code-mixed data in multiple language scripts [8].

Indian languages in general, and Dravidian languages like Tamil and Malayalam in particular, are considered as low-resource languages because of which data collection and annotation for any application becomes difficult [9]. The collection and annotation of code-mixed data for any application is intensified by the scarcity of resources. Further, the algorithms which process mono-lingual texts may not perform better for code-mixed texts. Hence, detecting sarcasm in code-mixed texts necessitates the development of multilingual language models and algorithms that help to process the code-mixed text which exhibits linguistic variations of more than one languages [10].

To address the challenges of processing code-mixed texts in Tamil and Malayalam for sarcasm identification, in this paper, we - team MUCS, describe the models submitted to the shared task ”Sarcasm Identification of Dravidian Languages (Malayalam and Tamil)” in DravidianCodeMix @FIRE 2023 [11]. Sarcasm identification problem is modeled as a binary classification task, to identify the given Malayalam and Tamil text as either ’Sarcastic’ or ’Non-sarcastic’. Three distinct binary classification models: i) Ensemble of ML classifiers (LR, RF, and SVC) trained with Term Frequency-Inverse Document Frequency (TF-IDF) of syllable n-grams ii) DL based

model (CNN trained with Keras embeddings), and iii) TL based models (transformer model trained with mBert and mDistilBert for Malayalam and Tamil respectively), are proposed to identify sarcasm in the given Malayalam and Tamil texts. [12]. As the given datasets are imbalanced, Text Augmentation (TA) approaches are explored to balance the Train set with the aim of improving the performance of the classifiers. Sample Tamil and Malayalam comments from the dataset along with their English translations are shown in Table 1.

Table 1
Sample Tamil and Malayalam comments from the dataset along with their English translations

Languages	Comments	English Translations	Labels
Tamil	இது செல்வராகவன் எனும் பைத்தியத்தின் மாபெரும் படைப்பு!!! Best of luck ' A selvaragavan flim'	This is the masterpiece of a madman called Selvaraghavan!!! Best of luck 'A selvaragavan flim'	Sarcastic
	உசிலம்பட்டி தேவர் சமுதாயம் சார்பாக .. வெற்றி பெற வாழ்த்துக்கள்	On behalf of the Usilampatti Devar community, congratulations on your success	Non-sarcastic
Malayalam	ശ്ശ ...!! അനു ചേച്ചി വിത്ത് മുലക്കച്ച പ്രതീക്ഷിച്ചു	Shhh...!! Anu sister was expecting seed milk	Sarcastic
	നീയൊക്കെ ആരാധിച്ചു ആരാധിച്ചു വളർത്തിയിട്ടല്ല ഞാൻ നടനായത്	I did not become an actor because you worshiped and raised me.	Non-sarcastic

The rest of the paper is organized as follows: Section 2 contains related work, Section 3 describes the methodology and Section 4 describes experiments and results followed by the conclusion and future work in Section 5.

2. Related Work

In spite of the availability of several models for sarcasm detection in code-mixed texts in some Indian languages [13], code-mixed texts in Dravidian languages are not yet explored fully in this direction and few of the relevant works are described below:

Kalaivani and Thenmozhi [14] implemented several ML models, DL model (Recurrent Neural Network with Long Short Term Memory (RNN-LSTM)) and TL-based model (transformer based classifier with BERT), for identifying sarcasm in English text obtained from Twitter and Reddit forums. They trained ML models with Doc2Vec vectors and TF-IDF of word unigrams, DL model with Keras embeddings and TL based model with BERT features. 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. Patil, Pravin K and Kolhe, SR [15] created manually annotated MarathiSarc - a Marathi dataset with 2,400 Marathi tweets for identifying sarcasm and implemented many ML models trained with TF-IDF of word unigrams. Among the ML models they experimented, XGBoost model outperformed the other models with a macro F1 score of 0.65. Pandey and Singh [16] trained several ML classifiers with TF-IDF of word unigrams, DL models (Deep Neural Network (DNN) CNN, LSTM) with keras embeddings, and TL-based model with BERT, for identifying sarcasm in code-mixed Hindi text. Further, they implemented a hybrid model by stacking LSTM network at the final layer of BERT model and their hybrid model obtained a remarkable macro F1 score of 0.98. Kumar et al. [17] implemented a DL model called sAtt-

BiLSTM convNet - a soft attention-based bidirectional LSTM model stacked on CNN to detect sarcasm in English text. Using Global Vectors (GLoVe) for semantic representation of words, their proposed model achieved a remarkable accuracy of 97.87%.

The dataset may be imbalanced and learning models trained on this imbalanced dataset may give results favoring the majority class, affecting the performance of the classifier. Several researchers have addressed the issue of data imbalance and some of the prominent ones are described below:

Abdullah et al. [18] highlighted the significance of resampling techniques to address data imbalance issues by adjusting the proportion of majority and minority instances either by over-sampling or under-sampling. Lee et al. [19] explored back-translation using Google translate[1] for TA to address the data imbalance issue for sarcasm detection in English text. A fine-tuned mBert model is presented by Kalaivani and Thenmozhi [20] for sentiment analysis in code-mixed Tamil, Malayalam and Kannada texts. As the given dataset is imbalanced, they augmented the Train set by employing transliteration and translation techniques and their proposed models obtained macro F1 scores of 0.603, 0.698, and 0.595 for Tamil, Malayalam, and Kannada texts respectively.

Ensemble approaches have shown improved performance in many text classification applications such as sentiment analysis, sarcasm detection, hate speech detection, etc. An ensemble of ML classifiers (SVC, LR, and RF) with soft voting considering TF-IDF of character n-grams features in the range (1, 3) is presented by Kumar et al. [21] for sentiment analysis in code-mixed Kannada, Malayalam, and Tamil texts. Their proposed models exhibited weighted F1 scores of 0.63, 0.73, and 0.62 for Kannada, Malayalam, and Tamil code-mixed texts respectively.

The related work highlights the research on sarcasm identification available in few Indian languages ensuring the need to develop efficient tools/models for sarcasm detection in other languages including the Dravidian languages Tamil and Malayalam.

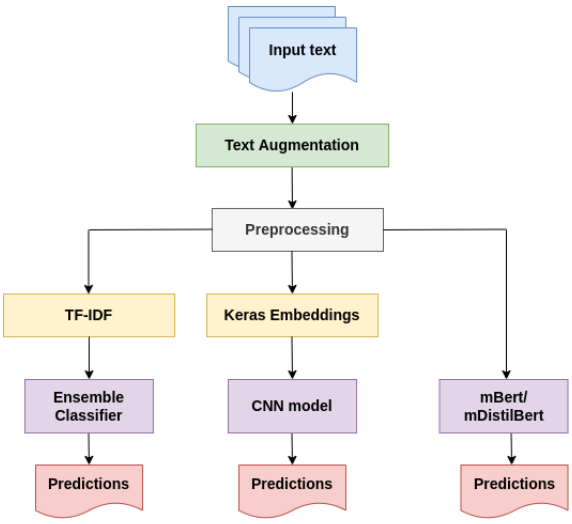

Figure 1: Framework of the proposed methodology

[1]https://translate.google.co.in/

Table 2
Distribution of Tamil and Malayalam code-mixed dataset

	Tamil		Malayalam	
Labels	Train set	Dev set	Train set	Dev set
Non_sarcastic	19,866	4,939	9,798	2,427
Sarcastic	7,170	1,820	2,259	588
Total	27,036	6,759	12,057	3,015

3. Methodology

The proposed methodology aims to identify sarcasm in Malayalam and Tamil code-mixed texts. Framework of proposed methodology is shown in Figure 1 and the components involved in the methodology are described below:

3.1. Text Augmentation

The datasets provided by the organizers for the shared task exhibit significant class imbalance as shown in Table 2. This data imbalance may affect the performance of the learning models as the training set is biased. Balancing the dataset either by the resampling techniques or by TA approaches can resolve the data imbalance issue. While resampling techniques increase the size of the minority class by replicating the text, TA approaches increase the size of the minority class by generating the diversified synthetic data. These approaches expand the training data to improve performance of the learning models for the Natural Language Processing (NLP) tasks, such as machine translation [22], text classification, question and answering system, etc., [23]. The statistics of the augmented datasets are shown in Table 3 and TA techniques used to increase the text belonging to minority class ('Sarastic') to balance the dataset are explained below:

1. Back-translation - is a technique where sentences from one language are translated into another language and then translated back to the original language. This technique generates textual data of distinct words for the original text while preserving the original context and meaning allowing a simple augmentation of text. Tamil and Malayalam text labeled as 'Sarcastic' is back-translated using Google translate for augmentation.
2. Prompt-based ChatGPT - has gained popularity in text generation [24]. Prompts serve as the primary means of interacting with ChatGPT that enables users to request information, generate content, or engage in conversations for a wide range of tasks including text generation, information retrieval, and translation. This work utilizes the prompt-based ChatGPT to generate synthetic data to increase the number of comments in Tamil dataset labeled as 'Sarcastic'.
3. Augmentation using the given dataset - the proposed models incorporate either language independent techniques (TF-IDF of syllable n-grams and keras embeddings) or multilingual models (mBert/mDistilBert pre-trained models) to obtain the features. Hence, adding the data of the same class from another dataset will augment the existing dataset. This is

Table 3
Statistics of Tamil and Malayalam comments belonging to 'Sarcastic' class after TA

TA Techniques	Tamil	Malayalam
Back-translation	2,251	2,251
Prompt based ChatGPT	1,339	-
Augmentation using given dataset	2,259	5,292
Total sarcastic comments after TA	5,849	7,543
Total sarcastic comments given in the dataset	**7,170**	**2,259**
Total sarcastic comments in the augmented dataset	**13,019**	**9,802**

carried out by adding Tamil sarcastic comments to Malayalam dataset and vice versa, to augment the given dataset.

3.2. Preprocessing

The objective of preprocessing is to clean the text with the intention of improving the accuracy of the learning models. In this direction, emojis are converted into text using demoji[2] library and punctuation, digits, and URLs are removed from the given Malayalam and Tamil code-mixed texts. English and Tamil stopwords available at Natural Language Toolkit (NLTK)[3] and github[4] are used as references to remove English and Tamil stopwords respectively, as they will not contribute to the sarcasm detection task.

3.3. Model Description

The preprocessed text is used to construct three distinct binary classification models: i) Ensemble of ML models, ii) DL model, and iii) TL-based models. The description of these models is given belows:

3.3.1. Ensemble of ML Models

Ensemble model consists of Feature Extraction and Classifier Construction as described below:
Feature Extraction - TF-IDF is a normalized representation of text documents used to reduce the impact of frequently occurring words across the documents. These vectors help to prioritize words that are distinctive to a document, making them valuable for various NLP tasks such as document retrieval, text classification, information retrieval etc. Syllables are distinct units of pronunciation with a single vowel sound. This representation is helpful in processing the text with non-romanized scripts as it provides meaningful tokens. In this work, syllable n-grams in the range (1, 3) are obtained from the preprocessed data and are vectorized using TfidfVectorizer[5]. Table 4 shows the sample Tamil and Malayalam code-mixed comments with their syllables and syllable unigrams, bigrams and trigrams.

[2]https://pypi.org/project/demoji/
[3]https://pythonspot.com/nltk-stop-words/
[4]https://gist.github.com/arulrajnet/e82a5a331f78a5cc9b6d372df13a919c
[5]https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

Classifier Construction - an ensemble method is a way of creating a new classifier by combining several diversified baseline classifiers such that the weakness of one classifier is overcome by the strength of another classifier. The idea behind ensemble approach is to consider a classifier combination as one single classifier and to improve its performance as compared to the performance of individual classifiers.

Table 4
Sample Tamil and Malayalam code-mixed comments with their syllables

Languages	Comments	Syllables and syllable unigrams, bigrams, and trigrams
Tamil	திருமா இனி மேல் நீ குருமா	['தி', 'ரு', 'மா', 'இ', 'னி', 'மே', 'ல் ', 'நீ', 'கு', 'ரு', 'மா']
	எனது நண்பர்கள் நிறைய	unigrams = ['எ', 'ன', 'து'], ['ந', 'ண்ப', 'ர்க', 'ள்'], ['நி', 'றை', 'ய'] bigrams = ['என', 'னது'], ['நண்ப', 'ண்பர்க', 'ர்கள்'], ['நிறை', 'றைய'] trigrams = ['எனது'], ['நண்பர்க', 'ண்பர்கள்'], ['நிறைய']
Malayalam	എം ജയചന്ദ്രൻ ഒന്നു കാണാൻ പറ്റുമോ	['എം','ജ', 'യ', 'ച', 'ന്ദ്ര', 'ൻ', 'ഒ', 'ന്നു', 'കാ', 'ണാ', 'ൻ', 'പ', 'റ്റു', 'മോ']
	നീയൊക്കെ ആരാധിച്ചു	unigrams = ['നീ', 'യൊ', 'ക്കെ'], ['ആ', 'രാ', 'ധി', 'ച്ചു'] bigrams = ['നീയൊ', 'യൊക്കെ'], ['ആരാ', 'രാധി', 'ധിച്ചു'] trigrams = ['നീയൊക്കെ'], ['ആരാധി', 'രാധിച്ചു']

Since the ensemble model incorporates multiple classifiers, it relies on a voting mechanism to predict class labels for unlabeled samples and hence, this approach is also known as a voting classifier. It allows the ensemble model to make accurate predictions for the new unlabeled samples by aggregating the decisions of individual classifiers. An ensemble of three ML classifiers, namely: LR, SVC, and RF, with hard voting is employed to detect sarcasm in the given texts.

- Logistic Regression - combines features linearly and uses regularization techniques to prevent over-fitting and the logistic function to classify instances into one of the predefined classes [25].
- Support Vector Classifier - excels in identifying intricate, non-linear relationships among the features, making it highly accurate in categorizing text documents [26].
- Random Forest - makes use of ensemble learning by combining a number of decision tree classifiers creating a ”forest” that is trained via bagging or bootstrap aggregation [27].

The hyperparameters of the ML classifiers are used with their default values.

3.3.2. Deep Learning Model

CNN architecture is employed to perform sarcasm identification in Malayalam and Tamil code-mixed texts. The model includes an embedding layer obtained from keras embeddings with a vocabulary of size 35,000 and embedding dimension 1000. It includes a convolutional layer featuring 64 filters and a kernel of size 2. For downsampling, the approach employs max pooling and subsequently the feature maps are transformed into a flattened representation as a one-dimensional vector. Eventually, an LSTM layer with 100 units is stacked with the previous layers to capture long-range dependencies and improving the model's ability to understand the context and meaning of the text. The final classification probabilities are generated through a dense layer employing the softmax activation function [28].

Table 5
Hyperparameters and their values used in mBert and mDistilBERT

Hyperparameters	mBert	mDistilBert
Layers	12	6
Dimensions	768	768
Attention heads	12	12
Learning Rate	1e-5	2e-5
Batch Size	32	16
Maximum Sequence Length	128	512
Dropout	0.3	0.3

3.3.3. Transfer Learning-based Models

TL is a learning approach where the knowledge acquired in training a source model is used to build another but related target model. Rather than building a model from scratch, this process accelerates learning and enhances the effectiveness of the target model by utilizing the pre-trained knowledge from the source task. This study makes use of two pre-trained models: mBert and mDistilBert, for Malayalam and Tamil code-mixed texts respectively, to detect sarcasm. Descriptions about mBert and mDistilBert models are as follows:

- mBert - is a BERT variant pre-trained on a vast amount of text data encompassing over 104 languages including Tamil, Malayalam and English in their native scripts, making it a multilingual language model by capturing and encoding semantic information from diverse linguistic contexts. It provides tokenizers and pre-trained embeddings for each token.
- mDistilBert - is a distilled version of BERT model pre-trained on a vast amount of text data encompassing over 104 languages including Malayalam and Tamil text extracted from Wikipedia along with their native and romanized scripts.

In this work, mBert model is fine-tuned on the Malayalam dataset, as majority of the comments in this dataset are in its native script along with the English text in Roman script and mDistilBert is fine-tuned on Tamil dataset since most of the comments are romanized. Hyperparameters and their values used to configure mBert and mDistilBert models are shown in Table 5.

4. Experiments and Results

The datasets provided for the shared task encompasses YouTube comments for both Malayalam and Tamil code-mixed texts. The majority of the comments are composed in native or Roman scripts, featuring code-mixed Tamil/Malayalam text with English.

Several experiments are conducted with different learning approaches and the models which gave the better results on the Development (Dev) set are evaluated on the Test set to get the predictions. The predicted labels of the Test set are evaluated by the organizers based on macro F1 scores and the performance of the proposed models on the Development and Test set are shown in the Table 6. From the table, it is clear that ensemble of ML classifiers performed

Table 6
Results of the proposed models in terms of macro F1 score

Malayalam				Tamil		
Models	**TA techniques**	**Dev set**	**Test set**	**TA techniques**	**Dev set**	**Test set**
Ensemble	Original data	0.68	0.68	Original data	0.68	0.67
				Original data +Back translation	0.66	0.67
	Original data +Back translation	0.71	**0.71**	Original data +Back translation +ChatGPT	0.70	0.68
	Original data +Back translation +Tamil	0.71	**0.71**	Original data +Back translation +ChatGPT +Malayalam	0.71	**0.70**
CNN	Original data	0.71	0.50	Original data	0.50	0.45
				Original data +Back translation	0.50	0.45
	Original data +Back translation	0.69	0.50	Original data +Back translation +ChatGPT	0.50	0.45
	Original data +Back translation +Tamil	0.68	0.64	Original data +Back translation +ChatGPT +Malayalam	0.50	0.45
mBert/ mDistilBert	Original data	0.45	0.45	Original data	0.71	0.69
				Original data +Back translation	0.68	0.68
	Original data +Back translation	0.69	0.69	Original data +Back translation +ChatGPT	0.70	0.69
	Original data +Back translation +Tamil	0.68	0.64	Original data +Back translation +ChatGPT +Malayalam	0.70	0.69

better with macro F1 scores of 0.71 and 0.70 securing 4^{th} and 5^{th} ranks for Malayalam and Tamil code-mixed texts respectively. This may be due to utilization of syllable n-grams features, which capture the meaningful tokens, particularly in processing non-roman script languages. Additionally, the pretrained models employed in this study have not effectively captured the sarcastic language nuances due to the limitations of the training data. The proposed models have shown improved macro F1 scores after TA except for Tamil code-mixed data which shows no improvement at all using CNN model.

5. Conclusion

This paper describes three distinct binary classification models: i) Ensemble of ML classifiers (LR, RF, and SVC) trained with TF-IDF of syllable n-grams in the range (1, 3), ii) DL model (CNN trained on Keras embeddings), and iii) TL based models (transformer model trained with mBert and mDistilBert for Malayalam and Tamil code-mixed texts respectively), submitted to "Sarcasm Identification of Dravidian Languages (Malayalam and Tamil)" in DravidianCodeMix@FIRE 2023 shared task. As the given dataset is imbalanced, TA techniques are employed to augment the data and the augmented data is used to train the proposed models. The proposed ensemble model exhibited macro F1 scores of 0.71 and 0.70 securing 4^{th} and 5^{th} ranks for Malayalam and Tamil code-mixed texts respectively.

References

[1] F. Balouchzahi, S. Butt, A. Hegde, N. Ashraf, H. Shashirekha, G. Sidorov, A. Gelbukh, Overview of CoLI-Kanglish: Word Level Language Identification in Code-mixed Kannada-English Texts at ICON 2022, in: Proceedings of the 19th International Conference on Natural Language Processing (ICON): Shared Task on Word Level Language Identification in Code-mixed Kannada-English Texts, 2022, pp. 38–45.

[2] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, N. Jose, S. Suryawanshi, E. Sherly, J. P. McCrae, Dravidiancodemix: Sentiment Analysis and Offensive Language Identification Dataset for Dravidian Languages in Code-Mixed Text, in: Language Resources and Evaluation, Springer, 2022, pp. 765–806.

[3] 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.

[4] A. Hegde, S. Lakshmaiah, Mucs@ mixmt: Indictrans-Based Machine Translation for Hinglish Text, in: Proceedings of the Seventh Conference on Machine Translation (WMT), 2022, pp. 1131–1135.

[5] A. Hegde, S. Coelho, H. Shashirekha, MUCS@ DravidianLangTech@ ACL2022: Ensemble of Logistic Regression Penalties to Identify Emotions in Tamil Text, in: Proceedings of the second workshop on speech and language technologies for Dravidian languages, 2022, pp. 145–150.

[6] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, S. Suryawanshi, N. Jose, E. Sherly, J. P. McCrae, Overview of the Track on Sentiment Analysis for Dravidian Languages in Code-mixed Text, in: Proceedings of the 12th Annual Meeting of the Forum for Information Retrieval Evaluation, 2020, pp. 21–24.

[7] N. Vasudevan, M. Haridas, P. Nedungadi, R. Raman, P. T. Daniels, D. L. Share, A Multi-Dimensional Framework for Characterizing the Role of Writing System Variation in Literacy Learning: A Case Study in Malayalam, Springer, 2023, pp. 1–34.

[8] A. Hegde, S. Banerjee, B. R. Chakravarthi, R. Priyadharshini, H. Shashirekha, J. P. McCrae, et al., Overview of the Shared Task on Machine Translation in Dravidian Languages, in: Proceedings of the second workshop on speech and language technologies for Dravidian languages, 2022, pp. 271–278.

[9] S. Swami, A. Khandelwal, V. Singh, S. S. Akhtar, M. Shrivastava, A Corpus of English-Hindi Code-Mixed Tweets for Sarcasm Detection, in: arXiv preprint arXiv:1805.11869, 2018.

[10] A. Shah, C. Maurya, How effective is incongruity? implications for code-mixed sarcasm detection, in: Proceedings of the 18th International Conference on Natural Language Processing (ICON), NLP Association of India (NLPAI), National Institute of Technology Silchar, Silchar, India, 2021, pp. 271–276. URL: https://aclanthology.org/2021.icon-main.32.

[11] 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.

[12] Y. Bai, B. Zhang, Y. Gu, T. Guan, Q. Shi, Automatic Detecting the Sentiment of Code-Mixed Text by Pre-training Model, in: Working Notes of FIRE, 2021.

[13] A. Kumar, S. R. Sangwan, A. K. Singh, G. Wadhwa, Hybrid Deep Learning Model for Sarcasm Detection in Indian Indigenous Language using Word-Emoji Embeddings, in: ACM Transactions on Asian and Low-Resource Language Information Processing, ACM New York, NY, 2023, pp. 1–20.

[14] A. Kalaivani, D. Thenmozhi, Sarcasm Identification and Detection in Conversion Context using BERT, in: Proceedings of the Second Workshop on Figurative Language Processing, 2020, pp. 72–76.

[15] Patil, Pravin K and Kolhe, SR, MarathiSarc: A Marathi Tweets Dataset for Automatic Sarcasm Detection of Marathi Tweets, 2022.

[16] R. Pandey, J. P. Singh, BERT-LSTM model for Sarcasm Detection in Code-mixed Social Media Post, in: Journal of Intelligent Information Systems, Springer, 2023, pp. 235–254.

[17] A. Kumar, S. R. Sangwan, A. Arora, A. Nayyar, M. Abdel-Basset, et al., Sarcasm Detection using Soft Attention-based Bidirectional Long Short-term Memory Model with Convolution Network, in: IEEE access, volume 7, IEEE, 2019, pp. 23319–23328.

[18] M. Abdullah, J. Khrais, S. Swedat, Transformer-Based Deep Learning for Sarcasm Detection with Imbalanced Dataset: Resampling Techniques with Downsampling and Augmentation, in: 2022 13th International Conference on Information and Communication Systems (ICICS), IEEE, 2022, pp. 294–300.

[19] H. Lee, Y. Yu, G. Kim, Augmenting Data for Sarcasm Detection with Unlabeled Conversation Context, in: Proceedings of the Second Workshop on Figurative Language Processing, Association for Computational Linguistics, Online, 2020, pp. 12–17. URL: https://aclanthology.org/2020.figlang-1.2. doi:10.18653/v1/2020.figlang-1.2.

[20] A. Kalaivani, D. Thenmozhi, Multilingual Sentiment Analysis in Tamil Malayalam and Kannada Code-Mixed Social Media Posts using MBERT, in: FIRE (Working Notes), 2021, pp. 1020–1028.

[21] A. Kumar, S. Saumya, J. P. Singh, An Ensemble-based Model for Sentiment Analysis of Dravidian Code-Mixed Social Media Posts, in: Working Notes of FIRE 2021-Forum for Information Retrieval Evaluation (Online). CEUR, 2021.

[22] A. Hegde, H. L. Shashirekha, KanSan: Kannada-Sanskrit Parallel Corpus Construction for Machine Translation, in: International Conference on Speech and Language Technologies for Low-resource Languages, Springer International Publishing Cham, 2022, pp. 3–18.

[23] C. Shorten, T. Khoshgoftaar, B. Furht, Text Data Augmentation for Deep Learning, in: Journal of Big Data, volume 8, 2021. doi:10.1186/s40537-021-00492-0.
[24] Q. Chen, H. Sun, H. Liu, Y. Jiang, T. Ran, X. Jin, X. Xiao, Z. Lin, Z. Niu, H. Chen, A Comprehensive Benchmark Study on Biomedical Text Generation and Mining with ChatGPT, in: bioRxiv, Cold Spring Harbor Laboratory, 2023, pp. 2023–04.
[25] J. S. Cramer, The Origins of Logistic Regression, Tinbergen Institute Working Paper, 2002.
[26] D. M. Tax, R. P. Duin, Support Vector Domain Description, in: Pattern recognition letters, volume 20, Elsevier, 1999, pp. 1191–1199.
[27] G. Biau, Analysis of a Random Forests Model, in: The Journal of Machine Learning Research, volume 13, JMLR. org, 2012, pp. 1063–1095.
[28] A. Hegde, F. Balouchzahi, K. G, S. Hosahalli Lakshmaiah, Trigger Detection in Social Media Text, 2023.