

Cross-Linguistic Sarcasm Detection in Tamil and Malayalam: A Multilingual Approach ^{*}

Dhanya Krishnan¹, Krithika Dharanikota² and B Bharathi³

^{1 2 3} Department of CSE, Sri Siva Subramaniya Nadar College of Engineering, Rajiv Gandhi Salai, Chennai, Tamil Nadu, India

Abstract

This paper introduces the methodologies developed for the Dravidian-CodeMix - FIRE 2023 task, focusing on the objective of classifying code-mixed comments in Tamil and Malayalam from YouTube videos as either Non-Sarcastic or Sarcastic. The research encompasses essential data preprocessing steps, experimentation employing diverse machine learning algorithms and feature vectors, and the comprehensive assessment of model performance. The Tamil model secured the second rank, and the Malayalam model secured the top position in the competition. Several methods, including the utilization of Count Vectorizer with MLP Classifier and Logistic Regression, as well as TF-IDF Vectorizer with MLP Classifier and TF-IDF Vectorizer with Random Forest Classifier, exhibited exceptional performance. Notably, validation accuracy for the Tamil dataset ranged from 0.72 to 0.78, accompanied by macro-averaged F1-scores spanning 0.73 to 0.77, while the Malayalam dataset showcased validation accuracy levels between 0.72 and 0.85, coupled with F1-scores spanning 0.60 to 0.77. These findings underscore the substantial potential of the employed techniques in addressing the intricate task of sarcasm detection while highlighting their pertinence in mitigating challenges related to linguistic diversity, code-mixing, and class disparities observed in online content across prominent languages. This research significantly contributes to the domain of cross-linguistic sarcasm detection and bears implications for multilingual sentiment analysis and the identification of cyberbullying within digital ecosystems.

Keywords

Sarcasm Detection, Sentiment Analysis, Dravidian Languages, Tamil, Malayalam

1. Introduction

Sarcasm refers to remarks, typically used in a humorous manner, to mock at, show irritation or convey contempt. It is a form of irony. With the recent surge in social media usage, sarcasm has played a vital role in conveying people's frustrations. It is found across a variety of platforms: Tweets, Instagram captions, and YouTube comments. However, due to the concealed nature of sentiment in sarcasm, identification and detection of the same becomes a tedious task. Sarcasm identification aids in varied applications— from sentiment analysis to detection of cyber-bullying, it becomes crucial to analyse sarcasm in texts. Tamil and Malayalam are Dravidian languages. Tamil is the official language of Tamil Nadu and Puducherry, Singapore and Sri Lanka, while Malayalam is the official language of Kerala and the union territory of Lakshadweep. Malayalam

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

[†]These authors contributed equally.

✉ dhanya2010402@ssn.edu.in (D. Krishnan); krithika2010087@ssn.edu.in (K. Dharanikota); bharathib@ssn.edu.in (B. Bharathi)



© 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)

is thought to be the closest major relative of Tamil, with the divergence of the two languages beginning in the 9th century AD. Owing to the inherent code-mixed nature characterising the usage patterns of these languages, the task of sentiment analysis emerges as a notably formidable undertaking. Traditional monolingual models often perform poorly on code-mixed datasets. This is attributed to the complexity demands of the code-switching that is required to analyse these datasets. The Dravidian-CodeMix - FIRE 2023 [10] shared task is formulated with the intent of encouraging collaborative research to devise suitable solutions for the discussed issue. The organisers presented the task of categorising code-mixed Tamil and Malayalam comments on YouTube videos featuring movie trailers as Non-Sarcastic and Sarcastic, aiming at analysing the sentiment polarities among viewers. This work presents the methodologies proposed for the Dravidian-CodeMix - FIRE 2023 task. It encompasses the utilisation of multiple models, with thorough analyses of their performances. Employed Machine Learning algorithms encompass Linear SVC, MLP Classifier, Naive Bayes, Random Forest, ULMFiT, KNN, and feature vectors such as TF IDF and CountVector.

The remainder of the paper is organised as follows: Section 2 discusses the literature survey of related works. The descriptions of data and the proposed methods are detailed in Sections 3 and 4 respectively. Section 5 underscores the achieved results and presents a thorough analysis of the performances of each model. Section 6 concludes the paper and contains a discussion about future research.

2. Related Work

The shared task [2] for Sarcasm detection using fuzzy logic suggests a unique ensemble approach based on text embedding that includes fuzzy evolutionary logic at the top layer. The model uses word and phrase embedding based on Word2Vec, GloVe, and BERT architecture. The datasets used were mainly social media datasets – Headlines dataset, the “Self-Annotated Reddit Corpus” (SARC), and the Twitter app dataset and the model was validated. Headlines dataset received the highest accuracy metric among the other datasets. The study regarding identification of Sarcasm in textual data [3] is a comprehensive review which covered articles on sarcasm detection published between 2008 and 2019 and elaborated on their classification approaches and reviewed their performance metrics, in an attempt to provide researchers with an invaluable insight into the research domain and to further improve sarcasm identification system using textual data. The survey also critically analyzed various data preparation (pre-processing) techniques and recent classification algorithms for sarcasm identification. In addition to this there is a study [4] which includes audio file detection as well as the textual data identification. This study uses a hybrid method that takes a combined vector of extracted audio and text features from specified models used in their study as the input. These combined features will compensate for the shortcomings of only text features and vice-versa. The results obtained from the hybrid model outperforms the results achieved when text and audio were detected individually : a total of 3 models were developed – one for textual data, one for audio features and lastly a hybrid model. Following this, we came across a study based on Sarcasm identification using machine learning algorithms in Twitter [5] which was aimed to improve the performance of the already accurate SVM and CNN-SVM algorithms for sarcasm detection. It showed that using lexical,

pragmatic, frequency, and part-of-speech tagging can contribute to the performance of SVM, whereas both lexical and personal features can enhance the performance of CNN-SVM. This study also recommends the use of two target labels when detecting the sarcastic statements tweets, which was implemented in our study.

Another study which proposed a simplified method to identify sarcasm within comments in a YouTube video channel of Bahasa Indonesia [6], which enables the channel or video owner to identify and remove or eliminate some comments that may arise hatred between the audience. Their results for the sarcasm comments using Regression Testing is claimed to be capable of identifying two types of sarcasm, which are propositional sarcasm and lexical sarcasm and further stating the performance metrics of Naïve Bayes to be very prominent in detecting the sarcasm in comments which criticized the current government for their plan to continue their power in their dataset. We also came across a shared task [7] which aims at developing a system that groups posts based on emotions, sentiment and find sarcastic posts if they are present datasets acquired from the lexical databases WordNet, SentiWordNet. They proposed a system which developed a prototype that helps to analyze the emotions of the posts namely anger, surprise, happy, fear, sorrow, trust, anticipation and disgust with three levels of severity. It also uses Sarcasm detection algorithms like Emoticon sarcasm detection, Hybrid sarcasm detection, Hashtag Processing, Interjection Word Start (IWT), in the end combining approaches of different methods like emotion detection, use of emoticons, patterns, etc. and identifies the social site comment is sarcastic or not. On the other hand [8], uses two ensemble based approaches - voted ensemble classifier and random forest classifier., which is contrasting to current approaches to sarcasm detection which rely on existing dataset of positive and negative sentiments for training the classifiers. They used a seeding algorithm to generate training dataset. Their proposed model also uses a pragmatic classifier to detect emoticon based sarcasm.

3. Experiment Data

The following section provides a detailed description of the data used in this study as well as the preprocessing techniques employed. The undertaken task has also been discussed comprehensively.

3.1. Data Description

The organisers presented participants with the task of analysing code-mixed Tamil and Malayalam text to identify and detect sarcasm. The dataset [9, 10] comprised user comments sourced from YouTube videos showcasing movie trailers. It was divided into two on the basis of language –Tamil and Malayalam. The textual content was code-mixed with the Roman script used alongside the traditional language. Code-mixing, in this context, refers to the simultaneous use of more than one language in a single text. The comments seamlessly integrated English along with the Dravidian language. The average sentence length of the corpora was specified as one by the organisers. Table 1 presents labelled comments for the Tamil and Malayalam datasets.

The Tamil dataset comprises a total of 27,036 training samples, with 7,170 instances labeled as sarcastic and 19,866 instances categorized as non-sarcastic. Similarly, the Malayalam dataset encompasses 9798 non-sarcastic comments and 2259 sarcastic comments, out of a total 12057

Table 1. Labelled Comments

| Language | Comment | Label |
|-----------|---|---------------|
| Tamil | காடுவெட்டியார் ஆசிரியுடன் படம் வெற்றி பெற வாழ்த்துக்கள் | Non-sarcastic |
| Tamil | Peeta parakkk thalivarr marana massss trailer Kolle ghandle irrke RAJINI ... DAAAA..... | Sarcastic |
| Malayalam | Underrated aakan pookunna adutha super malayalam padam!! | Non-sarcastic |
| Malayalam | Car jump pinne vere oru jump airilude... athu vendayirunnu | Sarcastic |

Table 2 Dataset statistics

| Language & Set | Sarcastic | Non-Sarcastic |
|----------------------------|-----------|---------------|
| Tamil - Training set | 7170 | 19866 |
| Malayalam - Training set | 2259 | 9798 |
| Tamil - Validation set | 1820 | 4939 |
| Malayalam - Validation set | 588 | 2427 |

training samples. There is an evident class imbalance in the data, mirroring real-world scenarios where such imbalances are commonly encountered. Table 2 depicts these statistics.

3.2. Task Description

The shared task of FIRE 2023 is to identify Sarcasm in text. It requires participants to develop a binary classification model to categorise code-mixed Tamil and Malayalam datasets as “Sarcastic” and “Non-Sarcastic”

Sarcastic: Comments which contain sarcastic phrases or instances, ie., comments whose true meaning diverges from what is written. Non-sarcastic: Comments which are direct and do not contain hidden intentions.

3.3. Data Pre-processing

The comments in the dataset consist of English alphabets as well as Tamil and Malayalam alphabets. Special characters and punctuations occur frequently in the dataset. The preprocessing step loops through a list of special characters and uses the str.replace() method to replace each

special character with a space in the text data of the training, validation, and test datasets as they were found to add no useful information to the text. Further, entries with missing values or labels were also removed from the dataset. The code first identifies columns in the datasets that are unnamed. The `dataset.columns.str.match("Unnamed")` line performs this check. Then, the `loc` function is used to select only those columns where the column names are not unnamed, thus removing unnecessary columns from the dataset.

Before pre-processing: Ithu 96' !! Anti aging icon Rajani Sir

After pre-processing: Ithu 96 Anti aging icon Rajani Sir

4. Proposed Methodology

This section details in-depth explanations for each of the experiments conducted for the shared task. Figure 1 depicts the general flow of the proposed methodologies.

4.1. Experiment 1 - Tamil: Count Vectorizer and MLP Classifier

For the first run, this work has used Count Vectorizer to extract features based on character-level-n-grams from the Tamil dataset. The length of the n-grams was declared as the range (1,3). In order to classify the extracted features as sarcastic or non-sarcastic, an MLP Classifier is employed. There exists one hidden layer with 128 neurons in this model. A maximum number of 10 iterations were performed to train the model on the training set. To avoid overfitting, the model was instructed to stop training if there is no improvement after 5 iterations. The trained classifier is then evaluated on the validation set with 5-fold cross-validation. The training and validation scores (Accuracy, Sensitivity, F1-score, Precision) are computed and recorded. This approach resulted in a validation accuracy of 0.78 and macro averaged f1-score of 0.73. The model was then used to predict the labels of the Tamil test set. The results released by the organisers shows a f1-score of 0.73 for the test set, ranking second among all participants.

4.2. Experiment 2 - Tamil: TF-IDF Vectorizer and MLP Classifier

In this experiment, the features have been extracted from the training and validation set using TF-IDF vectorizer based on character-level-n-grams where the length falls within the range (1,3). Following feature extraction, the MLP classifier is then trained on the training set. This classifier has a hidden layer consisting of 128 neurons. Similar to the first experiment, the model is limited to 10 iterations for training and 5 iterations if it shows no improvement. This is enforced to ensure there are no instances of overfitting. After successful training, the classifier is cross-validated with 5 folds on the validation set. The evaluation metrics are computed for both the training and evaluation set. It displayed a macro-averaged validation accuracy of 0.72 and f1-score of 0.77. The trained classifier was used to predict the headers for the comments in the test set.

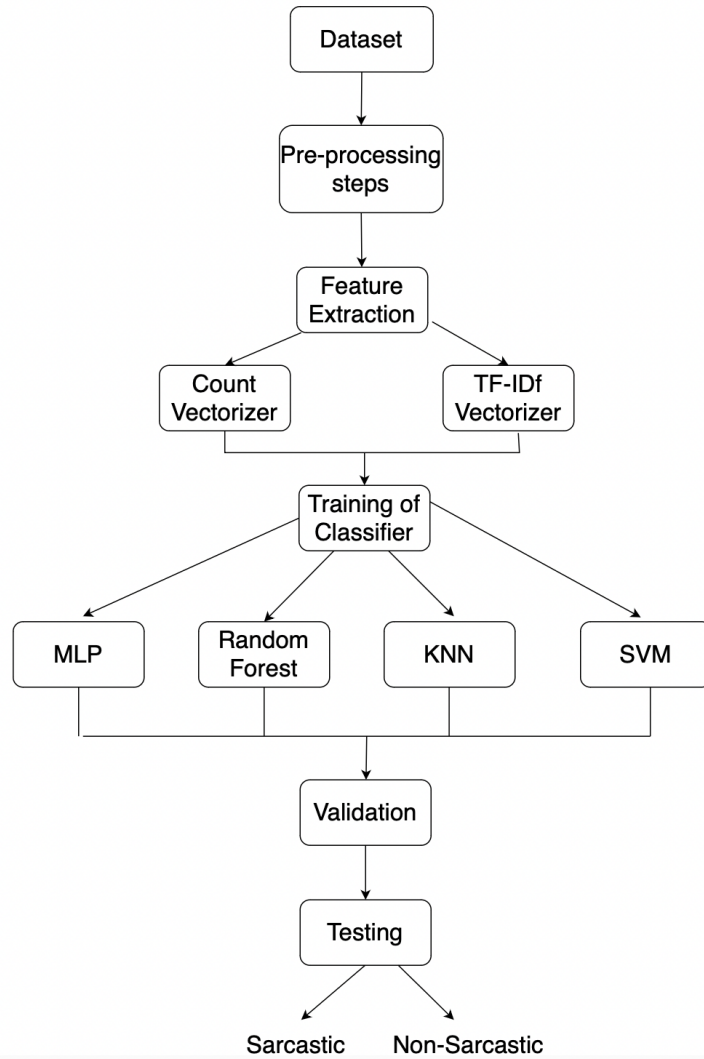


Fig.1. Process Flow

4.3. Experiment 3 - Tamil: TF-IDF Vectorizer and Random Forest Classifier

In order to combat the evident class imbalance, this approach uses oversampling. Firstly, the labels are converted to their numeric equivalents, with 1 representing “Sarcastic” and 0 representing “Non-Sarcastic”. Then, the features are extracted using TF-IDF vectorizer based on character-level-n-grams. To balance the class distribution, RandomOverSampler is used to oversample the “Sarcasm” class which has only 7170 samples as compared to 19866 samples in the “Non-Sarcastic” class. Following this, the Random Forest Classifier is initialised with 100 decision trees. The class weights are initialised as 0:1,1:3 so as to assign more weight to the minority class. The classifier is then trained on the resampled training data. After the RF classifier has been trained, the model is evaluated on the validation set and the metrics are calculated. This led to a macro-averaged validation f1-score of 0.72.

4.4. Experiment 4 - Malayalam: Count Vectorizer and MLP Classifier

To discriminate between sarcastic and non-sarcastic instances in the experiment on the Malayalam language dataset, the count vectorizer with MLP classifier approach that was utilized for the Tamil language dataset was applied. The Count Vectorizer approach was used throughout the feature extraction process to collect character-level n-grams, with a range of n-grams from 1 to 3. The statistics of the dataset showed the distribution, with 2259 occurrences in the training set classified as sarcastic and 9798 instances as non-sarcastic. In the following testing, 588 sarcastic instances and 2427 non-sarcastic instances were present in the validation set. The classification model was created using an MLP Classifier architecture, which had a solitary hidden layer with 128 neurons. A monitoring system was employed to stop training if no performance improvement was seen after 5 iterations, thereby reducing the danger of overfitting, and a maximum of 10 iterations were specified to enable optimal model training. The trained classifier was assessed against the validation set using a 5-fold cross-validation approach, and performance measures such as Accuracy, Sensitivity, F1-score, and Precision were cautiously recorded. This project produced outstanding results, with a validation accuracy of 0.85 and a weighted F1-score of 0.64. With a remarkable F1-score of 0.64, this model's proficiency was extended to the Malayalam test set, winning it the first spot in the competition standings.

4.5. Experiment 5 - Malayalam: Count Vectorizer with Logistic Regression

An alternate approach was used in a parallel analysis of the Malayalam language dataset to draw a distinction between sarcastic and non-sarcastic instances. The Count Vectorizer technique, which captures character-level n-grams with lengths ranging from 1 to 3, served as the basis for this strategy as well. The pattern of distribution was revealed by a statistical analysis of the dataset, with 9798 cases classified as non-sarcastic and 2259 instances assigned to the sarcastic class, making up the training set. And subsequently, 588 occurrences classified as sarcastic and 2427 instances classified as non-sarcastic were observed in the validation set. A Logistic Regression approach, which is recognized for its clarity and simplicity, was used to build the classification model. Through 5-fold cross-validation on the validation set, the model was rigorously evaluated after being trained and optimized for best performance. To evaluate the model's effectiveness, important metrics like Accuracy, Sensitivity, F1-score, and Precision were carefully monitored. The result was outstanding and demonstrated the model's proficiency with a validation accuracy of 0.84 and a weighted F1-score of 0.618.

4.6. Experiment 6 - Malayalam: TF-IDF and MLP Classifier

An entirely different approach was used in a separate investigation on the Malayalam language dataset to distinguish between sarcastic and non-sarcastic passages. The introduction of the Term Frequency-Inverse Document Frequency (TF-IDF) technique served as the cornerstone of this strategy and was important in capturing the essence of character-level n-grams with n-gram lengths ranging from 1 to 3. The training dataset, which was determined by a thorough analysis of the dataset statistics, had 2259 cases that were designated as sarcastic and 9798 examples that were non-sarcastic instances. The validation set resulted in 588 cases that were classified as sarcastic and 2427 instances that were non-sarcastic. The Multi-Layer (MLP)

Table 3. Metrics of Evaluation for above Experiments using validation dataset

| | Macro-Averaged Precision | Macro-Averaged Recall | Macro-Averaged F1-score | Macro-Averaged Accuracy |
|---------------------|---------------------------------|------------------------------|--------------------------------|--------------------------------|
| EXPERIMENT 1 | 0.7186 | 0.7442 | 0.7289 | 0.7186 |
| EXPERIMENT 2 | 0.7472 | 0.7232 | 0.7329 | 0.7232 |
| EXPERIMENT 3 | 0.715 | 0.715 | 0.715 | 0.720 |
| EXPERIMENT 4 | 0.7286 | 0.7581 | 0.7415 | 0.845 |
| EXPERIMENT 5 | 0.7112 | 0.7429 | 0.7252 | 0.8381 |
| EXPERIMENT 6 | 0.7785 | 0.7110 | 0.7357 | 0.8533 |

Classifier, recognized for its ability to capture complex correlations within data, helped to build the architecture of the classification model. The model was rigorously evaluated using 5-fold cross-validation on the validation dataset after careful training and deliberate fine-tuning. To evaluate the model’s efficacy, key performance measures including Accuracy, Sensitivity, F1-score, and Precision were closely recorded. The results were impressive, demonstrating the model’s skill with a validation accuracy of 0.85 and a weighted F1-score of 0.628.

5. Results and Discussion

Results from the cross-linguistic sarcasm detection experiments in Tamil and Malayalam were revelatory. Our team (SSNCSE1) secured high rankings as displayed in Table 5 and Table 6. Out of all the models we have experimented with, which are displayed in the Table 4, best-selected models demonstrated excellent performance using a variety of approaches, including Count Vectorizer with MLP Classifier and Logistic Regression, TF-IDF Vectorizer with MLP Classifier, and TF-IDF Vectorizer with Random Forest Classifier. The validation accuracy for the Tamil dataset ranged from 0.72 to 0.78, while the macro-averaged F1-score varied among the several studies from 0.73 to 0.77. The validation accuracy for the Malayalam dataset was similar, with values between 0.72 and 0.85 and F1-scores between 0.60 and 0.77. These results show how well the models are able to interpret the intricacies of sarcasm in code-mixed text comments sourced from YouTube videos that are both in Tamil and Malayalam. This work evaluates performance of the models using F1-score. F1 score is a harmonic mean of precision and recall - it is well-suited for evaluating sarcasm detection models because it balances both false positives and false negatives. In the context of sarcasm, where subtle linguistic nuances are crucial, F1-score provides a comprehensive measure, ensuring that the model’s ability to capture both sarcastic instances and non-sarcastic statements is effectively assessed, offering reliable performance evaluation.

The results show the complexity in the task of identifying sarcasm across languages and how significant it is in sentiment analysis. In spite of the major hurdles brought on by language variety, code-mixing, and class inequality, the suggested approaches showed encouraging potential. This study highlights the significance of handling language variances in order to get correct insights from content on the internet. It also sheds light on the multilingual complexities of sarcasm detection and provides the foundation for future developments in sentiment analysis such as restriction on posting a sarcastic comment under a video posted online. The same tendencies seen in the trials conducted in Tamil and Malayalam illustrate

Table 4. Weighted F1 scores of the machine learning models for evaluation dataset

| MODEL | WEIGHTED F1 SCORE |
|--|-------------------|
| Tamil Dataset | |
| SVM with Countvectorizer | 0.731 |
| MLP with Countvectorizer | 0.727 |
| KNN with Countvectorizer | 0.748 |
| Random Forest with Countvectorizer | 0.691 |
| SVM with TF-IDF Vectorizer | 0.78 |
| MLP with TF-IDF Vectorizer | 0.7746 |
| Random Forest with TF-IDF Vectorizer | 0.769 |
| Gradient Boosting | 0.66 |
| MLP with Indo-Aryan-XLM-R-Base - transformer model | 0.748 |
| UMLFIT | 0.764 |
| MNB | 0.73 |
| Malayalam Dataset | |
| SVM with Countvectorizer | 0.566 |
| MLP with Countvectorizer | 0.636 |
| KNN with Countvectorizer | 0.519 |
| Random Forest with Countvectorizer | 0.459 |
| Logistic Regression with Countvectorizer | 0.618 |
| SVM with TF-IDF Vectorizer | 0.361 |
| MLP with TF-IDF Vectorizer | 0.628 |
| KNN with TF-IDF Vectorizer | 0.572 |
| Logistic Regression with TF-IDF Vectorizer | 0.557 |

Table 5. Sarcasm identification task rank list for Tamil language

| Tamil | | | |
|-------|-----------------------|------|------|
| S.No | Team name | MF1 | Rank |
| 1 | hatealert_Tamil | 0.74 | 1 |
| 2 | ABC_tamil | 0.73 | 2 |
| 3 | SSNCSE1_Tamill | 0.73 | 2 |
| 4 | IRLabITBHU_tam | 0.72 | 3 |
| 5 | ramyasiva_tamil | 0.71 | 4 |
| 6 | ENDEAVOUR | 0.70 | 5 |
| 7 | MUCS_Tamil | 0.70 | 5 |
| 8 | Hydrangea_tamilrun3 | 0.69 | 6 |
| 9 | SSN_FeaturesAlpha_tam | 0.68 | 7 |
| 10 | YenCS_tam | 0.68 | 7 |
| 11 | TechWhiz_tam | 0.66 | 8 |

Table 6. Sarcasm identification task rank list for Malayalam language

| Malayalam | | | |
|-----------|-------------------------|------|------|
| S.No | Team name | MF1 | Rank |
| 1 | SSNCSE1_Malayalami | 0.74 | 1 |
| 2 | hatealert_Malayalam | 0.73 | 2 |
| 3 | ABC_malayalam | 0.72 | 3 |
| 4 | IRLabITBHU_mal | 0.72 | 3 |
| 5 | MUCS_mal | 0.71 | 4 |
| 6 | SSN_FeaturesAlpha_mal | 0.63 | 5 |
| 7 | TechWhiz_mal | 0.63 | 5 |
| 8 | YenCS_mal | 0.63 | 5 |
| 9 | Hydrangea_malayalamrun1 | 0.57 | 6 |
| 10 | ENDEAVOUR_malayalam | 0.53 | 7 |
| 11 | ramyasiva_malayalam | 0.52 | 8 |

the difficulties that code-mixing and linguistic traits present in both languages, pointing to the necessity of multilingual techniques to handle the same tasks in several languages.

6. Conclusion

In this work, we undertook a thorough investigation of cross-linguistic sarcasm detection in Tamil and Malayalam, two Dravidian languages. Count Vectorizer with MLP Classifier and Logistic Regression, TF-IDF Vectorizer with MLP Classifier, and TF-IDF Vectorizer with Random Forest Classifier are just a few of the numerous methodologies we used to tackle the challenging

task of sarcasm detection and classification in code-mixed text comments from YouTube videos.

Our findings not only demonstrated the ability of the suggested models to properly detect the hidden sentiment underlying sarcastic comments, but also brought to light the difficulties that multilingual sentiment analysis presents. The macro-averaged F1-scores and validation accuracies acquired across several tests showed favorable results, demonstrating the promise of our techniques in handling the complexity of sarcasm detection.

The study also emphasized the need of tackling issues like linguistic diversity, code-mixing, and class inequality because they are widespread in online content that is available in all major languages. Given that both Tamil and Malayalam showed analogous tendencies in our models, we underline the universality of these issues and the need of multilingual techniques for reliable sentiment analysis.

As a result, our study contributes to our knowledge of cross-linguistic sarcasm detection and establishes the groundwork for future studies in sentiment analysis. The results obtained have implications for practical applications such as sentiment-driven content analysis and the identification of cyberbullying in multilingual environments in addition to making contributions to the field of natural language processing. Our findings emphasizes the necessity for flexible and nuanced models capable of understanding the complexities as language nuances continue to impact online interactions.

References

- [1] Chakravarthi, Bharathi Raja, N Sripriya, B Bharathi, K Nandhini, Subalalitha Chinnadayar Navaneethakrishnan, Thenmozhi Durairaj, Rahul Ponnusamy, Prasanna Kumar Kumaresan, Kishore Kumar Ponnusamy, and Charmathi 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.
- [2] Sharma, D.K.; Singh, B.; Agarwal, S.; Pachauri, N.; Alhussan, A.A.; Abdallah, H.A. Sarcasm Detection over Social Media Platforms Using Hybrid Ensemble Model with Fuzzy Logic. *Electronics* 2023, 12, 937. URL:<https://doi.org/10.3390/electronics12040937>
- [3] Eke, C.I., Norman, A.A., Liyana Shuib et al. Sarcasm identification in textual data: systematic review, research challenges and open directions. *Artif Intell Rev* 53, 4215–4258 (2020). URL:<https://doi.org/10.1007/s10462-019-09791-8>
- [4] Santosh Kumar Bharti, Rajeev Kumar Gupta, Prashant Kumar Shukla, Wesam Atef Hatamleh, Hussam Tarazi, Stephen Jeswinde Nuagah, "Multimodal Sarcasm Detection: A Deep Learning Approach", *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 1653696, 10 pages, 2022. URL:<https://doi.org/10.1155/2022/1653696>
- [5] Sarsam, S. M., Al-Samarraie, H., Alzahrani, A. I., Wright, B. (2020). Sarcasm detection using machine learning algorithms in Twitter: A systematic review. *International Journal of Market Research*, 62(5), 578–598. URL:<https://doi.org/10.1177/1470785320921779>
- [6] W. Wijaya, I. M. Murwantara and A. R. Mitra, "A Simplified Method to Identify the Sarcastic Elements of Bahasa Indonesia in Youtube Comments," 2020 8th International Conference on Information and Communication Technology (ICoICT), Yogyakarta, Indonesia, 2020,

pp. 1-6, doi: 10.1109/ICoICT49345.2020.9166269. URL:<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=9166269isnumber=9166148>

- [7] S. Rendalkar and C. Chandankhede, "Sarcasm Detection of Online Comments Using Emotion Detection," 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2018, pp. 1244-1249, doi: 10.1109/ICIRCA.2018.8597368. URL:<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=8597368isnumber=8596764>
- [8] T. Jain, N. Agrawal, G. Goyal and N. Aggrawal, "Sarcasm detection of tweets: A comparative study," 2017 Tenth International Conference on Contemporary Computing (IC3), Noida, India, 2017, pp. 1-6, doi: 10.1109/IC3.2017.8284317. URL:<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=8284317isnumber=8284279>
- [9] Chakravarthi, B.R. Hope speech detection in YouTube comments. *Soc. Netw. Anal. Min.* **12**, 75 (2022). <https://doi.org/10.1007/s13278-022-00901-z>
- [10] Chakravarthi, B., Hande, A., Ponnusamy, R., Kumaresan, P., & Asoka Chakravarthi, R. (2022). 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, 100119. <https://doi.org/10.1016/j.jjime.2022.100119>