Sarcasm Detection and Identification of Dravidian Language Using Machine Learning Approach

Moogambigai A, Kamesh S, Paruvatha Priya B and Bharathi B

Department of Computer Science, Sri Sivasubramaniya Nadar College of Engineering, Chennai, India

Abstract

Sarcasm detection poses a significant challenge in sentiment analysis, especially on social media, where sarcasm is often used to convey opinions indirectly. This complexity is further exacerbated in multilingual settings, particularly with code-mixed languages like Tamil-English and Malayalam-English, where traditional sentiment analysis systems, typically trained on monolingual data, often fail due to the intricacies of code-switching at different linguistic levels. In this work, we develop an automated system for detecting sarcasm in code-mixed social media texts, with a focus on Tamil-English and Malayalam-English. We employ several machine learning classifiers, including Random Forest, Logistic Regression, Support Vector Machine, and Multinomial Naive Bayes, along with TF-IDF vectorization for feature extraction. Our system is trained on a newly developed gold standard corpus that reflects the real-world class imbalance typical of such datasets. The performance evaluation of the models shows that the Support Vector Machine and Random Forest classifiers achieve the highest accuracy, outperforming existing systems designed for monolingual sarcasm detection. These results represent a significant advancement in handling sarcasm in under-resourced languages and encourage further research into multilingual sentiment analysis in code-mixed contexts.

Keywords

Sarcasm detection, code-mixed languages, sentiment analysis, Tamil-English, Malayalam-English

1. Introduction

Sarcasm is a way of speaking where the actual meaning is different from the literal words. It's often used for irony, teasing, or humor, which makes it hard for sentiment analysis systems to detect. For example, if someone says, "Great job on breaking the vase!" The words seem positive, but the true meaning is negative. Detecting sarcasm is important to understand the real sentiment in text, especially on social media, [1]where tone and intent can be easily misread. In recent years, there has been a growing demand for effective sarcasm and sentiment detection systems tailored to the code-mixed texts prevalent on social media platforms. The Dravidian languages, Tamil and Malayalam, are widely spoken in South India and among the global diaspora. Tamil is recognized as an official language in India, Sri Lanka, and Singapore, while Malayalam is predominantly spoken in the Indian state of Kerala. However, the ease of typing in Roman script has led to widespread use of code-mixed Tamil-English and Malayalam-English in online communication. This paper introduces a new gold standard corpus for the detection of sarcasm and sentiment in code-mixed Tamil-English and Malayalam-English texts collected from social media [2]The task is to classify YouTube comments as either sarcastic or non-sarcastic at the message level. This shared task[3] represents the first known effort to address sarcasm detection in Dravidian code-mixed text, and it aims to foster research that illuminates how sarcasm is expressed in these multilingual scenarios.

2. Data Collection and Sources

For this task, we present a unique dataset composed of Tamil-English and Malayalam-English code-mixed sentences derived from YouTube video comments. The dataset is specifically curated to encompass all three types of code-mixed sentences: Inter-Sentential switch, Intra-Sentential switch, and Tag switching.

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

moogambigai2370071@ssn.edu.in (M. A); kamesh2370070@ssn.edu.in (K. S); paruvathapriya2370053@ssn.edu.in (P. P. B); bharathib@ssn.edu.in (B. B)

The comments in the dataset exhibit a mix of native script and Roman script, reflecting the linguistic complexity of code-mixing in social media [4]. Many comments are written in either Tamil or Malayalam script, combined with English lexicon or grammar, creating a rich, diverse dataset that captures various code-mixing patterns. Additionally, some comments are composed in Tamil or Malayalam script with English expressions interspersed, further enhancing the dataset's variety.

2.1. Dataset Structure

The dataset structure aligns with the shared task on sarcasm detection for Dravidian code-mixed languages [5], providing comprehensive training, development, and test sets for building and evaluating robust models. These datasets not only allow for the development of sarcasm detection systems but also serve as a resource for further research in code-mixed language processing and sentiment analysis in under-resourced languages.

Table 1 presents the dataset statistics for both Tamil-English and Malayalam-English code-mixed comments, indicating the number of samples in the training, development, and test sets. Both datasets present unique challenges due to their code-mixed nature. Social media comments often combine Tamil/Malayalam and English in a single sentence, sometimes even switching between languages within a single word. This characteristic increases the complexity of tokenization, feature extraction, and sarcasm detection, requiring specialized preprocessing steps. Additionally, the datasets reflect significant class imbalance, with sarcastic comments being much rarer than non-sarcastic ones.

Table 1
Dataset Statistics for Tamil-English and Malayalam-English code-mixed comments.

Language	Training Data	Development Data	Test Data
Tamil-English	29571	6337	6339
Malayalam-English	13189	2827	2827

2.1.1. Tamil-English:

The Tamil-English dataset is the larger of the two, containing 29,571 samples in the training set, 6,337 in the development set, and 6,339 in the test set. These comments are code-mixed between Tamil and English, where the majority of the text is written in Roman script. The dataset covers a wide range of sentiment expressions, including sarcasm, positive, neutral, and negative tones.Training Set: Comprises 29,571 comments that are used to train the machine learning models. This set includes labeled data that identifies whether a comment is sarcastic or not.Development Set: This set contains 6,337 labeled comments and is used for model validation and hyperparameter tuning during training. The development set plays a crucial role in preventing overfitting by allowing iterative testing.Test Set: The test set, with 6,339 comments, is used for the final evaluation of the models. These comments are provided without labels, and the model's predictions on this set determine its generalization performance.

2.1.2. Malayalam-English:

The Malayalam-English dataset is smaller in size but poses similar challenges due to code-mixing. It contains 13,189 samples in the training set, 2,827 in the development set, and 2,827 in the test set.Training Set: Consists of 13,189 code-mixed Malayalam-English comments, used to train the models. Like the Tamil-English dataset, this set also contains labeled data for sarcasm detection.Development Set: With 2,827 labeled comments, this set is used for evaluating and refining the model during training. Its smaller size compared to Tamil-English indicates that fewer resources are available for Malayalam, highlighting it as an under-resourced language.Test Set: The test set contains 2,827 comments. Similar to the Tamil-English dataset, this set is unlabeled and serves as the benchmark for testing the model's performance.

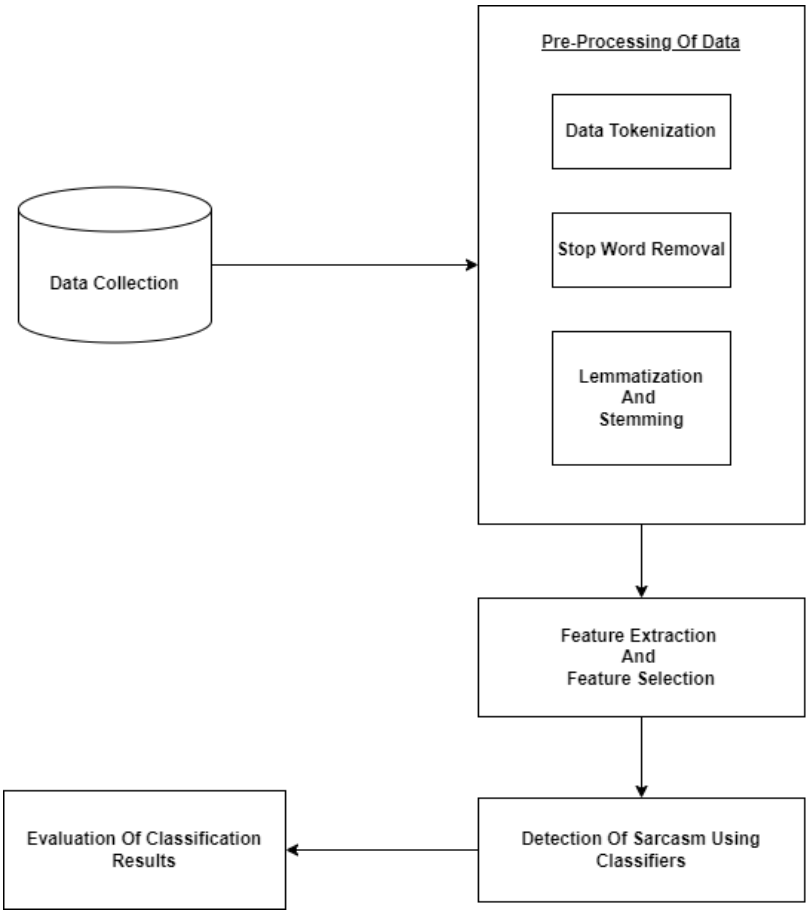

Figure 1: Framework of Proposed Methodology

3. Methodology

The framework of proposed methodology is shown in Figure 1

3.1. Classification Techniques for Malayalam Dataset

In our case, we tried several machine learning [6] models to classify sarcasm in the Malayalam-English dataset. Each model was selected based on its ability to handle the unique characteristics of the code-mixed data.

3.1.1. Logistic Regression:

Logistic Regression provided a straightforward approach to sarcasm detection. It performed well in identifying non-sarcastic comments, with a precision of 0.85, recall of 0.98, and an F1-score of 0.91. However, the model struggled with sarcastic comments, achieving a lower precision of 0.72, recall of 0.23, and F1-score of 0.35. The overall accuracy was 0.84, with a macro average F1-score of 0.63.

3.1.2. Random Forest:

The Random Forest model [7], an ensemble of decision trees, offered a balanced performance. It achieved an accuracy of 0.82, precision of 0.79, recall of 0.82, and an F1-score of 0.79. This model effectively managed the complexity of code-mixed text, providing consistent results across various metrics.

3.1.3. Support Vector Machine (SVM):

SVM [8], known for its robustness in text classification, performed slightly better than the Random Forest model. It achieved an accuracy of 0.83, precision of 0.81, recall of 0.83, and an F1-score of 0.81, making it a strong contender in sarcasm detection.

3.2. Classification Techniques for Tamil Dataset

For the Tamil-English dataset, we applied a range of classification techniques similar to those used in the Malayalam-English dataset. Logistic Regression showed mixed results [4], whereas Random Forest and SVM demonstrated more stable performances, aligning with the findings in the Malayalam-English dataset.

3.2.1. Logistic Regression:

The Logistic Regression model [9] provided a robust approach, excelling in non-sarcastic comment detection with a precision of 0.85, recall of 0.97, and F1-score of 0.91 during training. However, it struggled with sarcastic comments, especially in the testing phase, where it achieved a precision of 0.71, recall of 0.42, and F1-score of 0.53. The overall accuracy on the testing data was 0.80.

3.2.2. Random Forest with TF-IDF:

The Random Forest model, combined with TF-IDF vectorization, demonstrated strong performance during training, but its testing accuracy dropped to 0.78. It was effective in identifying non-sarcastic comments but faced challenges with sarcasm, resulting in a lower F1-score for sarcastic comments.

3.2.3. Multinomial Naive Bayes:

This model [10] provided balanced performance, achieving an accuracy of 0.84 during training and 0.76 on the testing data. It was particularly strong in detecting non-sarcastic comments but had limitations in identifying sarcastic ones.

3.2.4. TF-IDF Vectorizers:

TF-IDF Vectorization [11], while nearly perfect during training, faced generalization challenges on testing data, achieving an accuracy of 0.74. The model was effective for non-sarcastic comments but limited in detecting sarcasm.

3.3. Preprocessing Steps

Given the complexity of code-mixed text, several preprocessing steps were undertaken to prepare the data for modeling: Given the complexity of code-mixed text, several preprocessing steps were undertaken to prepare the dataset for modeling. First, text normalization was performed to standardize informal language, including slang, abbreviations, and non-standard spellings, thereby reducing noise in the data. Following normalization, tokenization was applied to split the comments into individual words or phrases, an essential step for feature extraction. Special care was taken in handling mixed-language tokens, as the dataset contained a combination of Tamil/Malayalam and English words. In cases where comments were written in native scripts, script conversion was employed to translate these into Roman script, ensuring consistency across the dataset, which predominantly consisted of Romanized text. This step was crucial for uniform text processing, particularly for code-mixed languages. A significant challenge encountered was the class imbalance in the dataset, where sarcastic comments were vastly outnumbered by non-sarcastic ones. To address this, various strategies were implemented, including oversampling, where the number of sarcastic samples was increased by duplicating existing ones, and undersampling, where the number of non-sarcastic samples was reduced to balance the classes.

Additionally, synthetic data generation techniques like SMOTE (Synthetic Minority Over-sampling Technique) were utilized to generate synthetic sarcastic comments, further mitigating the imbalance. Lastly, feature extraction was performed using TF-IDF vectorizers, which emphasize the importance of rare words that might carry sarcastic meanings. This approach enabled the system to capture the subtle nuances of sarcasm in code-mixed text, facilitating more accurate predictions.

3.4. Challenges Faced

During the training process, we encountered several challenges:

3.4.1. Class Imbalance:

As mentioned, the imbalance [12] between sarcastic and non-sarcastic comments posed a significant challenge. Despite applying techniques like oversampling and SMOTE, achieving a balanced dataset while maintaining the integrity of the data was difficult.

3.4.2. Code-Mixing Complexity:

The mixture of Tamil/Malayalam [13] with English within single comments introduced additional complexity. The variations in script, syntax, and linguistic structures made it challenging to develop models that could accurately capture the context and intent of the text.

3.4.3. Ambiguity in Annotation:

Annotating sarcasm [14] is inherently subjective, and even among native speakers, there can be differences in interpretation. Ensuring consistent and accurate labeling of the training data was a critical but challenging task.

3.4.4. Training and Testing without Labels:

While the training dataset [15] was annotated, the testing dataset provided for this study did not include labels, which added an extra layer of difficulty in evaluating model performance.

4. EXPERIMENTS AND RESULTS

4.1. Logistic Regression:

For both Tamil and Malayalam datasets, we implemented logistic regression models. These models are particularly effective in binary classification tasks such as sarcasm detection. We used advanced optimization techniques to handle the large and diverse datasets, ensuring that the models converge properly even with the intricate nature of code-mixed text.

4.2. Random Forest:

We employed Random Forest classifiers due to their robustness and ability to handle overfitting, making them ideal for our unevenly distributed datasets. The ensemble method, which builds multiple decision trees and merges them to get a more accurate and stable prediction, proved to be very effective, especially with the Tamil dataset using TF-IDF vectorization to enhance feature extraction.

4.3. Support Vector Machine (SVM):

The SVM models were configured with a linear kernel to capitalize on their strength in high-dimensional spaces, which is typical in text classification tasks like ours where features (words or terms) can be numerous. This model was applied to the Malayalam dataset and tuned to balance precision and recall, optimizing for the unique challenges of sarcasm detection in this language.

4.4. Multinomial Naive Bayes:

For the Tamil dataset, we utilized Multinomial Naive Bayes classifiers. These models are well-suited for discrete feature classification and were particularly chosen for their efficiency in handling large volumes of text data. The assumption that features follow a multinomial distribution aligns well with the word frequencies in code-mixed social media comments, aiding in better sarcasm identification.Each model was meticulously tuned to address the specific challenges posed by the sarcasm detection task, taking into account the complexities of code-switching, emotive content, and linguistic subtleties unique to Tamil and Malayalam code-mixed text. The performance of these models indicates a promising direction for further research in sarcasm detection within under-resourced languages.

4.5. Performance on Malayalam Validation Set

Table 2
Performance of Models on Malayalam Validation Set

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.84	0.85	0.98	0.91
Random Forest	0.82	0.79	0.82	0.79
Support Vector Machine (SVM)	0.83	0.81	0.83	0.81

Table 2. The Logistic Regression model achieved the highest performance with an accuracy of 0.84 and an F1-score of 0.91, indicating that it effectively captured both sarcastic and non-sarcastic posts. It also exhibited a high recall of 0.98, showing the model's ability to correctly identify most sarcastic instances, though this could reflect a bias towards recall over precision. The Random Forest model achieved 0.82 accuracy but with a relatively lower F1-score (0.79), which indicates slightly imbalanced performance in capturing sarcastic posts. The SVM model performed similarly to Random Forest, with an accuracy of 0.83 and F1-score of 0.81, showing consistent performance but without reaching the same level as Logistic Regression.

4.6. Performance on Tamil Validation Set

Table 3
Performance of Models on Tamil Validation Set

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.80	0.81	0.94	0.87
Random Forest with TF-IDF	0.78	0.79	0.95	0.86
Multinomial Naive Bayes	0.76	0.76	0.98	0.70

Table 3 shows the performance of various models on the Tamil validation set. Similar to the Malayalam set, Logistic Regression performed well with 0.80 accuracy and an *F1-score of 0.87. It balanced precision (0.81) and recall (0.94) effectively, indicating robustness in identifying sarcasm with relatively few false positives. The Random Forest with TF-IDF approach achieved 0.78 accuracy and an F1-score of 0.86, closely trailing Logistic Regression. Its high recall (0.95) shows a tendency towards minimizing false negatives, making it suitable for scenarios where missing sarcastic comments could be costly. The Multinomial Naive Bayes model performed slightly lower, with an accuracy of 0.76 and F1-score of 0.70. Although it achieved the highest recall (0.98) among the models, its precision was limited, indicating that it might over-classify sarcasm, resulting in more false positives.

5. Conclusion

In this study, we explored the challenging task of sarcasm detection in code-mixed Dravidian languages, specifically Tamil-English and Malayalam-English text from social media. By creating and analyzing a

unique dataset of YouTube comments, we highlighted the complexities of code-mixing, including Inter-Sentential, Intra-Sentential, and Tag switching, and their impact on sarcasm detection.Our approach involved selecting and extracting features that capture the linguistic and contextual subtleties of the text. By leveraging lexical, syntactic, sentiment, and code-mixing specific features, along with advanced contextual embeddings [16], we aimed to develop a robust model for identifying sarcasm in these mixed-language scenarios.For the Malayalam dataset, we employed Logistic Regression, Random Forest, and Support Vector Machine (SVM) as our classification techniques. Similarly, for the Tamil dataset, we utilized Logistic Regression, Random Forest with TF-IDF, and Multinomial Naïve Bayes approaches.The results demonstrate that integrating various feature types significantly improves sarcasm detection accuracy. However, the task remains difficult due to the intricate nature of sarcasm and the diversity of code-mixed languages. Class imbalance and variability in code-switching patterns further complicate the task, underscoring the need for more research in this area. Overall, this study contributes to the growing body of work on sentiment analysis in under-resourced languages, emphasizing the importance of context and specialized models when dealing with code-mixed languages. We submitted three models each for Tamil and Malayalam datasets, achieving notable placements in the evaluations, securing 5th place in Tamil and 9th place in Malayalam evaluations, reflecting our models' competitiveness and effectiveness in sarcasm detection within these challenging linguistic contexts.

6. Declaration on Generative AI

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

References

[1] A. Joshi, P. Bhattacharyya, M. J. Carman, Automatic sarcasm detection: A survey, ACM Computing Surveys (CSUR) 50 (2017) 1–22.

[2] P. Verma, N. Shukla, A. Shukla, Techniques of sarcasm detection: A review, in: 2021 international conference on advance computing and innovative technologies in engineering (ICACITE), IEEE, 2021, pp. 968–972.

[3] B. R. Chakravarthi, S. N, B. B, N. K, T. Durairaj, R. Ponnusamy, P. K. Kumaresan, K. K. Ponnusamy, C. Rajkumar, Overview of sarcasm identification of dravidian languages in dravidiancodemix@fire-2024, in: Forum of Information Retrieval and Evaluation FIRE - 2024, DAIICT , Gandhinagar, 2024.

[4] P. Liu, W. Chen, G. Ou, T. Wang, D. Yang, K. Lei, Sarcasm detection in social media based on imbalanced classification, in: Web-Age Information Management: 15th International Conference, WAIM 2014, Macau, China, June 16-18, 2014. Proceedings 15, Springer, 2014, pp. 459–471.

[5] R. A. Bagate, R. Suguna, Sarcasm detection of tweets without# sarcasm: data science approach, Indonesian Journal of Electrical Engineering and Computer Science 23 (2021) 993–1001.

[6] 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.

[7] H. Elgabry, S. Attia, A. Abdel-Rahman, A. Abdel-Ate, S. Girgis, A contextual word embedding for arabic sarcasm detection with random forests, in: Proceedings of the Sixth Arabic Natural Language Processing Workshop, 2021, pp. 340–344.

[8] A. Garg, N. Duhan, Sarcasm detection on twitter data using support vector machine., ICTACT Journal on Soft Computing 10 (2020).

[9] Y. Wang, A multinomial logistic regression modeling approach for anomaly intrusion detection, Computers & Security 24 (2005) 662–674.

[10] K. Sarkar, M. Bhowmick, Sentiment polarity detection in bengali tweets using multinomial naïve bayes and support vector machines, in: 2017 IEEE Calcutta Conference (CALCON), IEEE, 2017, pp. 31–36.

[11] S. Mihi, B. Ait Ben Ali, I. El Bazi, S. Arezki, N. Laachfoubi, Automatic sarcasm detection in dialectal arabic using bert and tf-idf, in: The Proceedings of the International Conference on Smart City Applications, Springer, 2021, pp. 837–847.
[12] A. Banerjee, M. Bhattacharjee, K. Ghosh, S. Chatterjee, Synthetic minority oversampling in addressing imbalanced sarcasm detection in social media, Multimedia Tools and Applications 79 (2020) 35995–36031.
[13] S. Chanda, A. Mishra, S. Pal, Sarcasm detection in tamil and malayalam dravidian code-mixed text., in: FIRE (Working Notes), 2023, pp. 336–343.
[14] P. Chaudhari, C. Chandankhede, Literature survey of sarcasm detection, in: 2017 International conference on wireless communications, signal processing and networking (WiSPNET), IEEE, 2017, pp. 2041–2046.
[15] P. Goel, R. Jain, A. Nayyar, S. Singhal, M. Srivastava, Sarcasm detection using deep learning and ensemble learning, Multimedia Tools and Applications 81 (2022) 43229–43252.
[16] N. Babanejad, H. Davoudi, A. An, M. Papagelis, Affective and contextual embedding for sarcasm detection, in: Proceedings of the 28th international conference on computational linguistics, 2020, pp. 225–243.