A Hybrid Approach to Sarcasm Detection in Dravidian Code-Mixed Texts

Prasun Maity[1,*,†], Dipanjan Saha[2,†], Sayan Das[3,†], Sainik Kumar Mahata[4,†] and Dipnakar Das[5,†]

[1]Haldia Institute of Technology
[2]Jadavpur University
[3]National Institute of Technology, Agartala
[4]Institute of Engineering & Management, Kolkata, University of Engineering and Management, Kolkata
[5]Jadavpur University

Abstract

This study presents a novel approach to Sarcasm Detection in Dravidian Code-mixed languages, specifically Tamil-English and Malayalam-English. Recognizing the challenges posed by code-mixing and the subtleties of sarcasm, we introduce a hybrid model that combines Convolutional Neural Networks (CNNs), Bidirectional Long Short-Term Memory (BiLSTM) networks, and AdaBoost. Trained on the dataset from the DravidianCodeMix@FIRE-2024 shared task, our model demonstrates the efficacy of integrating deep learning-based feature extraction with classical machine learning techniques for sarcasm detection in a multilingual, code-mixed context.

Keywords

Machine Learning, Deep Learning, Sarcasm Detection, Codemixing

1. Introduction

Sarcasm detection represents one of the most intricate challenges in sentiment analysis, where the conveyed meaning often starkly contrasts with the literal interpretation of the text [1]. This task becomes particularly complex in code-mixed environments, where users frequently blend multiple languages in their communication [2]. Code-mixing, the phenomenon where two or more languages are intertwined within a single sentence or conversation, is prevalent in multilingual communities, especially on social media platforms [3]. This linguistic interplay often utilizes non-native scripts, such as the Roman alphabet, for easier typing, further complicating the sentiment analysis process [4]. Traditional sentiment analysis models, predominantly designed and trained on monolingual data, often falter when applied to these intricate code-mixed texts due to their inability to handle the linguistic diversity and the nuanced use of sarcasm [5].

In this study, we focus on sarcasm detection within Dravidian code-mixed languages, specifically Tamil-English and Malayalam-English. These languages, widely spoken in southern India and by global diaspora communities, introduce unique linguistic complexities [6]. Tamil, with its origins in one of the oldest classical languages, and Malayalam, characterized by its alpha-syllabic script, present significant challenges in text processing, especially when interwoven with English [7]. The integration of English, a language with a vastly different structure and script, into these Dravidian languages results in code-mixed texts that are challenging for conventional sentiment analysis systems to accurately interpret [8].

Our approach involves the development of a robust model designed to accurately detect sarcasm in these code-mixed scenarios, addressing the limitations of existing sentiment analysis systems in multilingual and code-mixed environments [9]. By leveraging a hybrid model architecture that combines Convolutional Neural Networks (CNNs) with Bidirectional Long Short-Term Memory (BiLSTM)

Forum for Information Retrieval Evaluation, December 12-15, 2024, India
*Corresponding author.
†These authors contributed equally.
maitypro402@gmail.com (P. Maity); sahadipanjan6@gmail.com (D. Saha); sayandas20032727@gmail.com (S. Das); sainik.mahata@gmail.com (S. K. Mahata); dipankar.dipnil2005@gmail.com (D. Das)

networks, we aim to capture the intricacies of sarcasm in a multilingual context [10]. Furthermore, the integration of classical machine learning techniques, such as AdaBoost, enhances the model's ability to classify sarcastic content in these complex linguistic settings [11].

The dataset utilized in this study is sourced from the DravidianCodeMix@FIRE-2024 shared task, comprising code-mixed comments and posts from social media platforms like YouTube [12]. Annotated for sarcasm at the comment or post level, this dataset underscores the challenges posed by real-world data, including class imbalance and the brevity of social media texts [13]. The unique characteristics of this dataset necessitate innovative approaches to sarcasm detection, prompting the development of our hybrid model [14].

Through this research, we aim to contribute to the growing body of knowledge on sarcasm detection in multilingual and code-mixed environments [15]. Our findings have implications not only for improving sentiment analysis systems but also for enhancing the broader field of natural language processing (NLP), particularly in the context of increasingly global and linguistically diverse digital communication.

2. Dataset Description

The dataset for this study, provided by the *DravidianCodeMix@FIRE-2024* Shared Task, includes code-mixed comments and posts in Tamil-English and Malayalam-English. The data, sourced from social media platforms like YouTube, is annotated for sarcasm at the comment/post level, creating a message-level classification task. The dataset exhibits class imbalance, a common scenario in real-world data, with an average sentence length of one.

Table 1
Statistics of the Dataset

# of instances	Tamil	Malayalam
Non-sarcastic	1862	914
Sarcastic	673	217
TOTAL	2535	1131

This distribution highlights the imbalance and the brevity of the comments, typical of social media data.

3. Methodology

3.1. Data Preprocessing

The dataset is loaded from an Excel file with Tamil text data and labels. Text data (TEXT) and labels (LABELS) are extracted as features (X) and target (y). Labels are encoded numerically using LabelEncoder. Text is tokenized using Keras's Tokenizer, converting sentences into sequences of integers. Sequences are padded to a maximum length of 100 tokens. The data is split into training (80%) and testing (20%) sets.

3.2. Model Architecture

The model is a hybrid architecture that combines a *Convolutional Neural Network* (CNN) with a *Bidirectional Long Short-Term Memory* (Bi-LSTM) network. The architecture consists of the following layers:

- **Embedding Layer**: Converts the input sequences into dense vectors of fixed size (128 dimensions), where each word is represented by a dense vector of the embedding dimension.
- **Convolutional Layer**: Applies 128 filters with a kernel size of 5 to capture local patterns in the text, followed by a ReLU activation function.

- **MaxPooling Layer**: Reduces the dimensionality of the feature maps by selecting the maximum value from each pool, which helps in down-sampling the input.
- **Bidirectional LSTM Layer**: Applies an LSTM layer in both forward and backward directions to capture dependencies in the sequence data. The *return_sequences = True* parameter ensures that the output is a sequence, which is used in subsequent layers.
- **Global Max Pooling Layer**: Aggregates the maximum value from each feature map output by the LSTM, reducing the dimensionality and preserving important features.
- **Dense Layer**: Fully-connected layer with 128 units and ReLU activation, responsible for further feature extraction and non-linear transformations.
- **Dropout Layer**: Applies a 50% dropout to prevent overfitting by randomly setting some of the inputs to zero during training.
- **Output Layer**: A dense layer with a single unit and sigmoid activation function, providing the final binary classification output (sarcastic or non-sarcastic).

3.3. Traning

The Keras model is trained using the *Adam* optimizer and *binary cross-entropy* loss function, with accuracy as the evaluation metric. The training process includes *early stopping mechanism* to avoid overfitting by monitoring the validation loss, and model checkpoints to save the best-performing model. The model is trained for up to 20 epochs with a batch size of 32, and 20% of the training data is used as a validation set.

After training, the model is used to extract features from both the training and testing data, which are then used for further classification.

3.4. Evaluation

The extracted features from the trained Keras model are used to train an AdaBoost classifier, with a Decision Tree as the base estimator. The AdaBoost model is configured with 50 estimators and utilizes the *SAMME* algorithm for boosting. The trained AdaBoost model is evaluated on the test set, with predictions compared against the true labels.

The evaluation metrics include a classification report displaying precision, recall, F1-score, and support for each class. Additionally, a confusion matrix is generated and visualized using a heatmap to provide insights into the model's performance, highlighting the correct and incorrect predictions across classes.

4. Results

Our hybrid model achieved competitive results, effectively distinguishing between sarcastic and non-sarcastic comments in both Tamil and Malayalam datasets. Please refer Table 2.

5. Conclusion

This study presents a robust approach to sarcasm detection in Dravidian Code-mixed texts, combining the strengths of CNN-BiLSTM for feature extraction with AdaBoost for classification. The model effectively addresses the challenges posed by code-switching and sarcasm, offering a promising solution for sentiment analysis in multilingual and code-mixed environments.

6. Future Work

The future direction of this research holds significant potential for further advancements in sarcasm detection, particularly in the context of Dravidian code-mixed languages like Tamil-English and Malayalam-English. One promising avenue is the incorporation of attention mechanisms within the existing model

Table 2
Performance of the model on both the datasets

	Precision	Recall	F1-Score
On TAMIL Dataset			
	Precision	**Recall**	**F1-Score**
Non-sarcastic	0.96	0.93	0.94
Sarcastic	0.88	0.79	0.83
Accuracy			0.92
On MALAYALAM Dataset			
	Precision	**Recall**	**F1-Score**
Non-sarcastic	0.97	0.98	0.97
Sarcastic	0.90	0.88	0.89
Accuracy			0.96

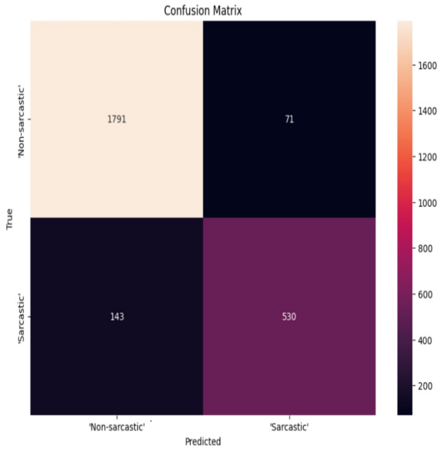

Figure 1: Confusion Matrix of TAMIL dataset Sarcasm Detection

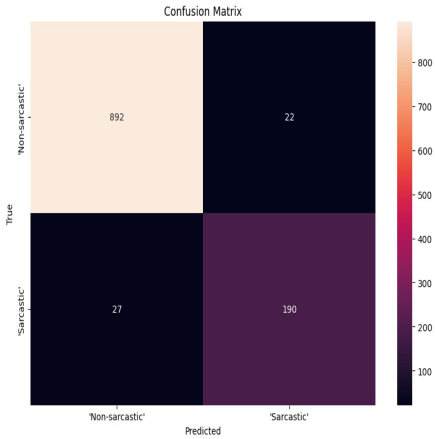

Figure 2: Confusion Matrix of MALAYALAM dataset Sarcasm Detection

architecture. Attention mechanisms have demonstrated their efficacy in various natural language processing tasks by allowing the model to focus on the most relevant parts of the input, thereby enhancing the model's interpretability and performance. Integrating attention layers could help the model better capture the nuances of sarcasm, which often relies on subtle cues spread across different parts of the sentence.

Moreover, the exploration of transformer-based models, such as BERT or GPT, tailored specifically for code-mixed languages, could provide a substantial leap in accuracy and robustness. These models, known for their deep contextual understanding and ability to handle complex language structures, could be fine-tuned on sarcasm detection tasks, offering a more sophisticated approach compared to traditional models.

Expanding the dataset to include a broader range of code-mixed languages beyond Tamil-English and Malayalam-English is another crucial step. Incorporating languages such as Hindi-English, Kannada-English, or other South Asian language pairs could make the model more versatile and applicable to a wider appearance. This would not only improve the model's generalizability but also contribute to the growing body of research on multilingual sarcasm detection.

Declaration on Generative AI

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

References

[1] Joshi, A., Bhattacharyya, P., & Carman, M. J. (2017). Automatic sarcasm detection: A survey. *ACM Computing Surveys (CSUR)*, 50(5), 1-22.

[2] Poria, S., Cambria, E., Hazarika, D., Majumder, N., Zadeh, A., & Morency, L. P. (2017). Context-dependent sentiment analysis in user-generated videos. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 873-883).

[3] Ghosh, A., Li, G., Veale, T., Rosso, P., Shutova, E., Barnden, J., & Reyes, A. (2015, June). SemEval-2015 task 11: Sentiment analysis of figurative language in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* (pp. 470-478).

[4] Ghosh, D., & Veale, T. (2016, June). Fracking sarcasm using neural network. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis* (pp. 161-169).

[5] Joshi, A., Tripathi, V., Patel, K., Bhattacharyya, P., & Carman, M. (2016, December). Are word embedding-based features useful for sarcasm detection? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 1006-1011).

[6] Poria, S., Hazarika, D., Majumder, N., Naik, G., Cambria, E., & Mihalcea, R. (2019). MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 527-536).

[7] Ghosh, A., & Veale, T. (2017). Magnets for sarcasm: Making sarcasm detection timely, contextual, and very personal. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 482-491).

[8] Ghosh, D., Guo, W., & Muresan, S. (2015, September). Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1003-1012).

[9] Hazarika, D., Poria, S., Gorantla, S., Cambria, E., Zimmermann, R., & Mihalcea, R. (2018, June). Cascade: Contextual sarcasm detection in online discussion forums. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 1837-1848).

[10] Joshi, A., Sharma, V., & Bhattacharyya, P. (2015, June). Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* (pp. 757-762).

[11] Bharti, S. K., Babu, K. S., & Jena, S. K. (2015). Parsing-based sarcasm sentiment recognition in Twitter data. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (pp. 1373-1380). IEEE.

[12] Amir, S., Wallace, B. C., Lyu, H., Carvalho, P., & Silva, M. J. (2016, June). Modelling context with user embeddings for sarcasm detection in social media. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning* (pp. 167-177). Association for Computational Linguistics.

[13] Bamman, D., & Smith, N. A. (2015, June). Contextualized sarcasm detection on Twitter. In *Proceedings of the 9th International Conference on Web and Social Media (ICWSM)* (pp. 574-577). Association for the Advancement of Artificial Intelligence.

[14] Rajadesingan, A., Zafarani, R., & Liu, H. (2015, February). Sarcasm detection on Twitter: A behavioral modeling approach. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining* (pp. 97-106). Association for Computing Machinery.

[15] Poria, S., Cambria, E., Hazarika, D., & Vij, P. (2016). A deeper look into sarcastic tweets using deep convolutional neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 1601-1612). International Committee on Computational Linguistics.