

# Leveraging Transfer Learning and Deep Recurrent Networks for Sarcasm Detection in Tamil Language Text

Kogilavani Shanmugavadivel<sup>1</sup>, Subhadevi K<sup>2</sup>, Sowbharanika Janani J S<sup>3</sup> and Rahul K<sup>4,\*</sup>

Department of AI, Kongu Engineering College, Perundurai, Erode

## Abstract

Our study uses advanced natural language processing (NLP) techniques to handle the difficulty of detecting sarcasm in Tamil text. The first step is to thoroughly clean and preprocess the data in order to remove any undesirable characters and standardize the text for analysis. The preprocessed data is tokenized and ready for usage in machine learning models. Three models are investigated: DistilBERT, GRU, and LSTM. DistilBERT, a lightweight but effective model, is ideal for detecting sarcasm because of its ability to capture minor contextual elements in text. It gets an F1 score of 0.74 on the test set, making it the best performer. A GRU-based model, constructed using PyTorch, is also designed to handle sequential text data, employing techniques such as dropout regularization and bidirectional layers to boost performance. Finally, the LSTM model developed in Keras is hyperparameter tuned to improve its capacity to identify irony in Tamil. Overall, the study demonstrates that various models, particularly DistilBERT, are excellent in detecting sarcasm in Tamil literature. This study emphasizes the need of tailored NLP algorithms for detecting sarcasm in individual languages and provides useful insights for future multilingual sentiment analysis research.

## Keywords

Natural Language Processing (NLP), DistilBERT, GRU (Gated Recurrent Unit), LSTM (Long Short-Term Memory), Sarcasm Detection, Sequence Classification

## 1. Introduction

Sarcasm identification became considerably more complex when working with code-mixed data, which combined elements from several languages. Due to the possibility of linguistic interactions altering tone and sentence structure, this phenomenon affected the identification of strange content. Parsing and interpreting sarcasm in Tamil-English code-mixed texts proved more challenging because of syntactic and lexical variations. By extracting and using particular linguistic traits to improve sarcasm detection in mixed-language contexts, [1], [2], and [3] showed the effectiveness of feature selection in addressing such complications.

Our study made use of a carefully chosen dataset of Tamil texts, comprising samples that were both sarcastic and non-sarcastic. To make sure it was suitable for training and testing sarcasm detection algorithms, this dataset was carefully produced. We employed a thorough preparation pipeline, which included tokenization and normalization, to clean and organize the data in order to enhance the performance of our models. Similar to the approach in [4], we extracted significant features from the text to prepare our dataset and improve the model's performance and accuracy.

We utilized DistilBERT, GRU, and LSTM, three different machine learning models, to tackle the problems associated with sarcasm detection. Each model brought unique benefits to the task. While GRU and LSTM models were built to handle sequential text data and detect long-term dependencies, which were crucial for recognizing sarcasm, DistilBERT, a condensed version of BERT, used pre-trained language representations to identify contextual variations [5]. We were able to employ different strategies to enhance sarcasm detection in Tamil by combining these models, showing the advantages of each tactic.

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

\*Corresponding author.

✉ kogilavani.sv@gmail.com (K. Shanmugavadivel); subhadevik.22aid@kongu.edu (S. K);  
sowbharanikajananijs.22aid@kongu.edu (S. J. J. S); rahulk.22aid@kongu.edu (R. K)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

## 2. Related Works

A systematic analysis of machine learning methods for sarcasm detection reveals that Support Vector Machines (SVM) are especially good at finding sarcasm in Twitter data, where pleasant statements can mask negative feelings. Through sophisticated semantic and behavioral labeling strategies, the combination of SVM and CNN improves accuracy [6].

In a different study, deep learning features from a CNN are combined with unique contextual data to detect sarcasm in tweets. According to the study, Logistic Regression exhibits superior performance in classifying these combined features, resulting in high values of F1-measure, accuracy, precision, and recall [7].

An alternative method uses Weka for classifier performance, TextBlob for polarity analysis and preprocessing, and RapidMiner for sentiment evaluation when working with Twitter data. Effectiveness of classifiers and sentiment analysis can be better understood by utilizing Naïve Bayes and SVM models [8].

Discrete manual features and continuous neural network features are both used in the study on neural networks with deep learning for sarcasm detection, which approaches the issue as a binary classification task. In comparison to manual approaches, it is found that bi-directional gated recurrent artificial neural networks and pooling networks greatly improve accuracy [9].

In a subsequent chapter, sarcasm detection is investigated using models that combine linguistic and pragmatic insights, offering a comparative study of machine learning classifiers. This illustrates how deep learning techniques may effectively grasp contextual differences [10].

Pre-trained models like BERT and RoBERTa are used in research on neurological sarcasm detection, and they incorporate context data from previous utterances. The top model performs well in the Sarcasm Shared Task 2020, achieving an F1 score of 0.790 [11].

An analysis of sarcasm detection techniques reveals that 50 percentage accuracy is only attained for Hindi text when Bag-of-Words features are combined with SVM. According to [12], this finding emphasizes the necessity for more sophisticated methods to enhance detection performance.

99 percentage accuracy for news headlines and 82 percentage accuracy for Reddit are achieved by an ensemble model that combines LSTM, GRU, and CNN with word embeddings such as fastText and Word2Vec. According to [13], this model operates more accurately and steadily than earlier models.

For sarcasm detection, the paper presents a multi-head attention-based BiLSTM model that outperforms conventional feature-rich SVM models by utilizing pragmatic, semantic, and lexical features to improve classification accuracy [14].

Using a clearer dataset of news headlines, the research provides a hybrid neural network with attention mechanisms, addressing the problem of noisy Twitter datasets. This method increases the accuracy of classifying sarcasm by about 5 percentage [15].

Lastly, a study of hybrid, deep learning models, and standard machine learning techniques for English sarcasm detection is presented, with an emphasis on utilizing pragmatic, semantic, and lexical features to increase classification accuracy [16].

## 3. Problem and System Description

This system's purpose was to detect sarcasm in comments that blended Tamil and English, which was difficult because the two languages were often switched within a single comment. Combining Tamil and English words made it even more difficult for the model to recognize sarcasm in addition to understanding the intended meaning.

Recent studies, including a shared task organized as part of the DravidianCodeMix effort, had brought attention to this issue. This collaborative endeavor examined the detection of sarcasm in the Dravidian languages of Tamil and Malayalam, highlighting the challenges associated with sarcasm recognition in code-mixed settings [17] [18] [19].

The approach employed a machine learning model that was trained on instances of both sarcastic and

non-sarcastic comments in order to address this. Recurrent networks and transfer learning, two deep learning approaches, were used to help the model find trends in the way users transitioned between languages in their comments. To enhance the system’s comprehension of the nuanced aspects of satirical Tamil-English remarks, pre-trained language models such as DistilBERT were also employed.

### 3.1. Dataset Description

The dataset consists of 29,570 rows of labels and text, displaying user comments from YouTube written in both Tamil and English. The text column contains code-switching, while the labels column assigns either sarcastic or non-sarcastic labels to each comment. This annotated dataset is useful for creating models that recognize sarcasm in mixed Tamil and English code, providing valuable training data.

The distribution of datasets among training, validation, and test sets is summarized in Table 1.

**Table 1**

Dataset Description

Dataset	No. of Comments
Train	29570
Validation	6336
Test	6338

A balanced approach to model evaluation and development is ensured by dividing the dataset into 29,570 comments for training, 6,636 for validation, and 6,338 for testing.

In addition, Table 2 shows a typical row containing code-mixed text and the label that goes with it, giving an example of the dataset structure.

**Table 2**

Sample Dataset

Text	Labels
All the best to the team....	Non-sarcastic
Apadilam nadakathu nadakavum koodathu..	Non-sarcastic
Chindhamani kolakkes.....like podraa	Sarcastic
Dai intha padatha eppada release pannuvangada	Sarcastic

## 4. Methodology

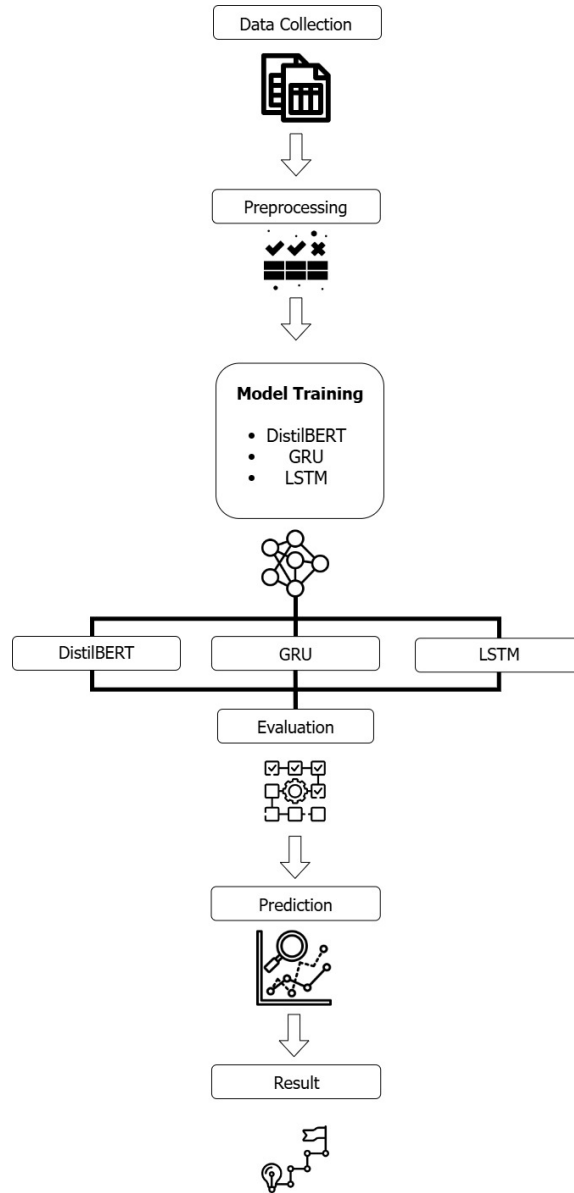
The following methodology outlines the steps involved in detecting sarcasm using DistilBERT, GRU, and LSTM models. The process encompasses three main components: diagrammatic representation, preprocessing steps, and algorithm explanation.

### 4.1. Diagrammatic Representation of Proposed Work

The figure 1 illustrates the entire process of sarcasm detection. The process begins with data collection, followed by preprocessing, model selection (DistilBERT, GRU, and LSTM), model training, evaluation, and finally prediction. This end-to-end process ensures that raw text data is processed, models are trained effectively, and predictions are made on unseen data.

### 4.2. Preprocessing Steps

In this stage, raw textual data is converted into a format suitable for model training. The first stage in preprocessing is text cleaning. To ensure consistency, the text is converted to lowercase, and all non-alphanumeric characters are deleted, leaving only letters, numbers, and a few punctuation symbols



**Figure 1:** Proposed system pipeline

such as exclamation points and periods, which may have semantic value in sarcasm identification. Furthermore, extra whitespace between words is removed in order to guarantee consistency in the input data.

Then, tokenization is performed. The DistilBERT model uses the Hugging Face library's AutoTokenizer for tokenization. This tokenizes the text into subword units while retaining padding and truncation to accommodate variable-length inputs. The GRU and LSTM models use similar tokenization procedures, but the text is tokenized into sequences of word indices that are then padded to a constant length. This phase additionally involves label encoding, which converts sarcastic and non-sarcastic labels into binary values (1 for sarcastic and 0 for non-sarcastic), preparing the data for supervised learning.

### 4.3. Predictions on Test Data

We apply the learned patterns to a fresh collection of statements in the Predictions on Test Data phase. Firstly, we utilize the tokenizer we created before to prepare the text and load the examine dataset. After the maximum period is chosen at some point throughout schooling, the sequences are padded to ensure

that they all have the same period. DistilBERT, GRU, and LSTM all depend their predictions entirely on these processed information. While the LSTM and GRU models concentrate on phrase knowledge, the DistilBERT version employs its superior architecture to explore linguistic styles. Once the predictions are made, the next step is to list the model outputs to determine whether or not each statement is a joke going forward. These results are transferred to another column in the test data set for explicit analysis. The quality in this section shows a good adaptation to new material, and shows the ability to recognize humor in unique texts. Overall, the results examine the applicability of the models to real-world, boundary-crossing emotional research, especially when dealing with complex language.

## 5. Result

The objective of this study was to create a sarcasm detection system that works well with deep learning techniques, specifically with DistilBERT, GRU, and LSTM models. A dataset comprising both sarcastic and non-sarcastic comments was used to train each model. The DistilBERT model demonstrated its efficacy in identifying sarcasm with a validation accuracy of 0.80 and a macro F1 score of 0.80. While the GRU classifier attained an accuracy and F1 score of 0.79, the LSTM model only managed to acquire an accuracy of 0.80 and a lower F1 score of 0.72 are shown in Table 3. The test dataset final findings, after being submitted to the CodaLab competition, produced an F1 score of 0.74. These results validate the models' capacity to identify sarcasm and demonstrate the need for additional development and research in this area.

**Table 3**

Validation Accuracy and Macro average F1 Score

Model	Accuracy	F1 score
DistilBERT	0.80	0.80
GRU	0.79	0.79
LSTM	0.80	0.72

## 6. Conclusion

The goal of this work was to apply machine learning models such as DistilBERT, GRU, and LSTM to identify criticism in Tamil YouTube comments. Our method was successful as we were able to obtain a macro F1 score of 0.71 by meticulously cleaning the data and using sophisticated neural network algorithms. Though it was not completely consistent, the model demonstrated good flexibility in response to various kinds of data. Enhancing user engagement and content management in social environments requires the ability to recognize subtle linguistic subtleties in Tamil, a difficult task that this research addresses and adds to the field of natural language processing. The study's findings, taken together, provide a framework for future research on language use across linguistic contexts.

## Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT in order to: drafting content, grammar and spelling check, etc. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

## References

- [1] B. R. Chakravarthi, A. Hande, R. Ponnusamy, P. K. Kumaresan, R. Priyadharshini, How can we detect homophobia and transphobia? experiments in a multilingual code-mixed setting for

- social media governance, *International Journal of Information Management Data Insights* 2 (2022) 100119.
- [2] N. Sripriya, T. Durairaj, K. Nandhini, B. Bharathi, K. K. Ponnusamy, C. Rajkumar, P. K. Kumaresan, R. Ponnusamy, C. Subalalitha, B. R. Chakravarthi, Findings of shared task on sarcasm identification in code-mixed dravidian languages, *FIRE 2023* 16 (2023) 22.
  - [3] B. R. Chakravarthi, Hope speech detection in youtube comments, *Social Network Analysis and Mining* 12 (2022) 75.
  - [4] M. S. M. Suhaimin, M. H. A. Hijazi, R. Alfred, F. Coenen, Natural language processing based features for sarcasm detection: An investigation using bilingual social media texts, in: 2017 8th International conference on information technology (ICIT), IEEE, 2017, pp. 703–709.
  - [5] M. Y. Manohar, P. Kulkarni, Improvement sarcasm analysis using nlp and corpus based approach, in: 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), 2017, pp. 618–622. doi:10.1109/ICCONS.2017.8250536.
  - [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] M. S. Razali, A. A. Halin, L. Ye, S. Doraisamy, N. M. Norowi, Sarcasm detection using deep learning with contextual features, *IEEE Access* 9 (2021) 68609–68618. doi:10.1109/ACCESS.2021.3076789.
  - [8] S. Saha, J. Yadav, P. Ranjan, Proposed approach for sarcasm detection in twitter, *Indian Journal of Science and Technology* 10 (2017) 1–8.
  - [9] M. Zhang, Y. Zhang, G. Fu, Tweet sarcasm detection using deep neural network, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: technical papers, 2016, pp. 2449–2460.
  - [10] N. Chatterjee, T. Aggarwal, R. Maheshwari, Sarcasm detection using deep learning-based techniques, *Deep Learning-Based Approaches for Sentiment Analysis* (2020) 237–258.
  - [11] N. Jaiswal, Neural sarcasm detection using conversation context, in: Proceedings of the second workshop on figurative language processing, 2020, pp. 77–82.
  - [12] A. D. Dave, N. P. Desai, A comprehensive study of classification techniques for sarcasm detection on textual data, in: 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), IEEE, 2016, pp. 1985–1991.
  - [13] 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.
  - [14] A. Kumar, V. T. Narapareddy, V. A. Srikanth, A. Malapati, L. B. M. Neti, Sarcasm detection using multi-head attention based bidirectional lstm, *Ieee Access* 8 (2020) 6388–6397.
  - [15] R. Misra, P. Arora, Sarcasm detection using hybrid neural network, *arXiv preprint arXiv:1908.07414* (2019).
  - [16] P. Katyayan, N. Joshi, Sarcasm detection approaches for english language, *Smart Techniques for a Smarter Planet: Towards Smarter Algorithms* (2019) 167–183.
  - [17] B. R. Chakravarthi, N. Sripriya, B. Bharathi, K. Nandhini, S. C. 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.
  - [18] 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.
  - [19] 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, DAICT*, Gandhinagar, 2024.