Hybrid Neural Network Model for Sentiment Analysis of Code-Mixed Dravidian Languages

Swapnil Raj
Roll No: 2024PGCSDS01
National Institute of Technology Jamshedpur
Under the guidance of

Dr. Jitesh Pradhan

Associate Professor
Department of Computer Science and Engineering
National Institute of Technology Jamshedpur

October 6, 2025

Abstract

Sentiment analysis for morphologically rich and code-mixed Dravidian languages presents unique challenges. This paper introduces a powerful hybrid neural architecture for feature extraction, combining **MuRIL-BERT embeddings**, **character-level CNNs**, a **multi-kernel word-level CNN**, and a **Bidirectional GRU with attention**, further augmented with statistical TF-IDF features. We evaluate this architecture using two classification strategies: an end-to-end approach with an MLP classifier and a two-stage pipeline where the neural network acts as a feature extractor for a Random Forest model.

On the Tamil dataset, the end-to-end model achieved a weighted F1-score of **0.6261**. The two-stage pipeline demonstrated excellent generalizability and success on the other languages, achieving a weighted F1-score of **0.7352** on the Malayalam dataset and **0.6215** on the Kannada dataset. These results validate that our hybrid feature extraction architecture provides a comprehensive representation for sentiment analysis and that the two-stage approach is a highly effective and adaptable strategy for this task in low-resource Dravidian languages.

1 Introduction

Sentiment analysis is a key task in Natural Language Processing (NLP) with applications in opinion mining, market research, and social media analytics. While transformer models such as BERT have shown remarkable success for high-resource languages, morphologically rich and code-mixed Dravidian languages such as Tamil, Malayalam, and Kannada present unique challenges. These include complex inflections, agglutination, spelling variations, and the frequent mixing with English in online text.

To address these challenges, we propose a **hybrid deep learning model** that serves as a powerful feature extractor, integrating contextual embeddings from MuRIL-BERT with local n-gram detectors, character-level morphological features, and sequential modeling. We then evaluate two distinct classification strategies on top of these rich features: a fully integrated end-to-end model and a flexible two-stage pipeline. We validate our methods on Tamil, Malayalam, and Kannada to demonstrate cross-lingual effectiveness and compare the different classification approaches.

2 Methodology

The architecture integrates four categories of features: contextual, morphological, sequential, and statistical. These components form a powerful feature extractor.

2.1 Contextual Embeddings (MuRIL-BERT)

We use Google's MuRIL-BERT (`google/muril-base-cased`), a multilingual transformer pre-trained on Indian languages. The contextual embeddings capture semantic information and code-switching. The $[CLS]$ token is used as a global sentence representation.

2.2 Character-level Features (CharCNN)

Dravidian languages often include spelling variations and agglutinative forms. We employ a character-level CNN where each token is represented as a sequence of characters embedded in a 50-dimensional space. Convolutions with kernel sizes 2, 3, and 4 capture morphological patterns.

2.3 Local and Sequential Modeling (CNN-BiGRU with Attention)

Token embeddings (concatenation of MuRIL and CharCNN outputs) are passed through a 1D CNN with multiple kernel sizes. These representations are then fed into a Bidirectional GRU to capture long-range dependencies. An attention mechanism computes a weighted sum of hidden states:

$$u_t = \tanh(W_a h_t + b_a)$$
$$\alpha_t = \frac{\exp(u_t^T v_a)}{\sum_{j=1}^{T} \exp(u_j^T v_a)}$$
$$c = \sum_{t=1}^{T} \alpha_t h_t$$

where h_t is the BiGRU hidden state at time t, and c is the attention-weighted sequence representation.

2.4 Statistical Features

- **TF-IDF Features:** Character-level TF-IDF vectors with n-gram range (1-6) ($d_{tfidf} = 5000$) are projected into a 64-dimensional dense space.
- **Auxiliary Features:** Handcrafted features such as token count, character count, and sentiment lexicon counts are projected into a 32-dimensional space.

2.5 Feature Fusion

The final representation is a concatenation of the BERT $[CLS]$ vector, attention-based BiGRU representation, TF-IDF projection, and auxiliary projection:

$$v_{final} = [v_{CLS}; c_{BiGRU}; f_{tfidf}; f_{aux}]$$

This rich feature vector serves as the input to the classification stage.

2.6 Classification Strategies

We evaluate two distinct strategies for the final classification step.

2.6.1 End-to-End MLP Classifier

The fused feature vector (v_{final}) is passed directly into a Multi-Layer Perceptron (MLP) head with ReLU activation and dropout for final classification. The entire network, from MuRIL to the MLP head, is trained jointly. This end-to-end approach was utilized for the Tamil sentiment analysis task.

2.6.2 Two-Stage Pipeline (Feature Extraction + RF Classifier)

Alternatively, the trained neural network can be used as a dedicated feature extractor. In this two-stage approach (also known as stacking), the process is:

1. **Stage 1:** The hybrid neural network is trained as described above.
2. **Stage 2:** The trained network is used to extract the final feature vectors (v_{final}) for the entire dataset. These vectors are then used to train a separate, robust classifier.

For the Malayalam and Kannada experiments, we employed a Random Forest classifier in this two-stage approach, which can offer greater stability and performance depending on the dataset characteristics.

3 Model Pipeline Diagram and Code

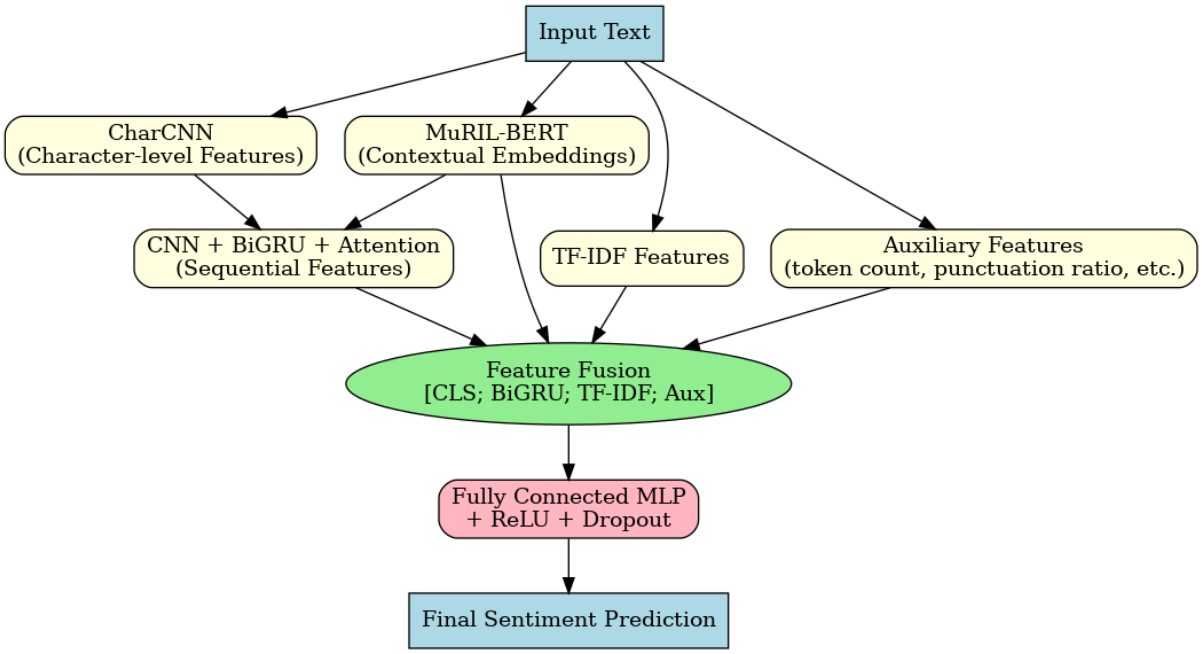

Figure 1: High-level overview of the Hybrid Pipeline.

4 Experiments

4.1 Datasets

We evaluate our approach on three Dravidian language datasets.

Tamil Dataset: We use the `tamil_sentiment_full.csv` dataset containing five sentiment classes, with the distribution shown in Table 1. Classes with fewer than three samples were discarded. The dataset was split into training (70%), validation (15%), and test (15%) and was evaluated using the end-to-end MLP classifier.

Table 1: Tamil Dataset Class Distribution

Sentiment Label	Count
Positive	24,871
unknown_state	7,004
Negative	5,339
Mixed_feelings	2,314
not-Tamil	2,087
Total	41,615

Malayalam Dataset: We use the `mal_full_sentiment.tsv` dataset, which comprises 19,616 samples (Table 2). The data was split into training (80%), validation (10%), and test (10%) sets and evaluated using the two-stage pipeline.

Kannada Dataset: We use the `kannada_sentiment.csv` dataset (Table 3). After removing classes with fewer than two samples, the data was split into training (80%), validation (10%), and test (10%) sets and evaluated using the two-stage pipeline.

Table 2: Malayalam Dataset Class Distribution

Sentiment Label	Count
Positive	7,907
unknown_state	6,502
Negative	2,600
not-malayalam	1,445
Mixed_feelings	1,162
Total	19,616

Table 3: Kannada Dataset Class Distribution

Sentiment Label	Count
Positive	3,518
Negative	1,484
not-Kannada	1,136
unknown_state	842
Mixed_feelings	691
Total	7,671

4.2 Training Details

Training was conducted for 4–6 epochs with batch size 32. We used the AdamW optimizer with differential learning rates (2×10^{-5} for BERT, 1×10^{-3} for the head). A focal loss with $\gamma = 2.0$ was used to mitigate class imbalance. Best checkpoints were saved based on validation macro F1. For the two-stage pipeline, the Random Forest used 200 estimators and a max depth of 10.

5 Results and Discussion

5.1 Tamil Sentiment Analysis Results (End-to-End)

The end-to-end model achieved the following performance on the Tamil test set:

- **Accuracy:** 64.18%
- **Macro F1:** 0.5161
- **Weighted F1:** 0.6261

The model is strongest on the *Positive* class (F1 = 0.7877) due to its larger sample size, while performance on *Mixed_feelings* is weaker (F1 = 0.2359), reflecting inherent ambiguity.

Table 4: Classification Report on Tamil Test Set (End-to-End MLP)

Class	Precision	Recall	F1-Score	Support
Mixed_feelings	0.3699	0.1732	0.2359	739
Negative	0.4319	0.4770	0.4533	784
Positive	0.7512	0.8279	0.7877	3731
not-Tamil	0.6886	0.6358	0.6611	313
unknown_state	0.4525	0.4324	0.4423	1036
Accuracy		0.6418		6603
Macro Avg	0.5388	0.5093	0.5161	6603
Weighted Avg	0.6208	0.6418	0.6261	6603

5.2 Malayalam Sentiment Analysis Results (Two-Stage)

The two-stage pipeline achieved strong performance on the Malayalam test set. The neural network feature extractor reached a best validation Macro F1-score of **0.6821**, and the final Random Forest classifier yielded:

- **Accuracy:** 74.03%
- **Weighted F1-score:** 0.7352

5.3 Kannada Sentiment Analysis Results (Two-Stage)

The two-stage approach was also effective for Kannada. The neural network feature extractor reached a best validation Macro F1-score of **0.5443**. The final Random Forest classifier achieved:

- **Accuracy:** 64.04%
- **Weighted F1:** 0.6215

Table 5: Classification Report on Malayalam Test Set (Two-Stage RF)

Class	Precision	Recall	F1-Score	Support
Mixed_feelings	0.5564	0.3190	0.4055	232
Negative	0.6131	0.5942	0.6035	520
Positive	0.7842	0.8180	0.8007	1582
not-malayalam	0.8154	0.8408	0.8279	289
unknown_state	0.7356	0.7571	0.7462	1301
Accuracy		0.7403		3924
Macro Avg	0.7010	0.6658	0.6768	3924
Weighted Avg	0.7343	0.7403	0.7352	3924

Table 6: Classification Report on Kannada Test Set (Two-Stage RF, n=1,535)

Class	Precision	Recall	F1-Score	Support
Mixed_feelings	0.4250	0.1232	0.1910	138
Negative	0.6553	0.6465	0.6508	297
Positive	0.6929	0.7884	0.7375	704
not-Kannada	0.5769	0.6608	0.6160	227
unknown_state	0.4894	0.4083	0.4452	169
Accuracy		0.6404		1535
Macro Avg	0.5679	0.5254	0.5281	1535
Weighted Avg	0.6220	0.6404	0.6215	1535

5.4 Comparative Analysis and Discussion

The consistent performance across all three languages demonstrates that our hybrid architecture effectively handles the challenges common to Dravidian languages, such as morphological richness (via CharCNN) and code-mixing (via MuRIL-BERT).

The superior performance of the two-stage pipeline on Malayalam is particularly noteworthy. This suggests that for certain datasets, decoupling feature extraction from classification can be beneficial. The Random Forest, a robust ensemble method, may be better at finding a generalizable decision boundary from the rich, static features provided by the neural network, compared to the end-to-end MLP which learns the features and classifier simultaneously. This highlights the flexibility of our architecture, allowing for the selection of the most effective classification strategy based on the specific language and dataset.

Table 7: Cross-Lingual Performance Comparison

Metric	**Tamil**	**Malayalam**	**Kannada**
Accuracy	64.18%	74.03%	64.04%
Weighted F1	0.6261	0.7352	0.6215
Macro F1	0.5161	0.6768	0.5281

6 Conclusion

We presented a hybrid feature extraction architecture using MuRIL-BERT, CharCNN, and a CNN-BiGRU pipeline for Dravidian language sentiment analysis. By evaluating two different classification strategies, we demonstrated the framework's flexibility and effectiveness. The end-to-end model achieved a weighted F1-score of 0.6261 on Tamil. The two-stage pipeline, which used the neural network as a feature extractor for a Random Forest classifier, proved highly successful, achieving weighted F1-scores of 0.7352 on Malayalam and 0.6215 on Kannada.

The cross-lingual validation confirms the generalizability of our feature extractor. The results establish our hybrid architecture as an effective framework for sentiment analysis in low-resource Dravidian languages, and they highlight the value of exploring different classification strategies to maximize performance for specific datasets. Future work will explore larger pretrained Indic transformers and advanced data augmentation techniques to further improve performance on challenging sentiment categories like *Mixed_feelings*.