MysticCIOL@DravidianLangTech 2025: A Hybrid Framework for Sentiment Analysis in Tamil and Tulu Using Fine-Tuned SBERT Embeddings and Custom MLP Architectures

Minhaz Chowdhury, Arnab Laskar, Taj Ahmad, Azmine Toushik Wasi[†]
Shahjalal University of Science and Technology, Sylhet, Bangladesh
[†]**Correspondence:** azmine32@student.sust.edu

Abstract

Sentiment analysis is a crucial NLP task used to analyze opinions in various domains, including marketing, politics, and social media. While transformer-based models like BERT and SBERT have significantly improved sentiment classification, their effectiveness in low-resource languages remains limited. Tamil and Tulu, despite their widespread use, suffer from data scarcity, dialectal variations, and code-mixing challenges, making sentiment analysis difficult. Existing methods rely on traditional classifiers or word embeddings, which struggle to generalize in these settings. To address this, we propose a hybrid framework that integrates fine-tuned SBERT embeddings with a Multi-Layer Perceptron (MLP) classifier, enhancing contextual representation and classification robustness. Our framework achieves validation F1-scores of 0.4218 for Tamil and 0.3935 for Tulu and test F1-scores of 0.4299 in Tamil and 0.1546 on Tulu, demonstrating its effectiveness. This research provides a scalable solution for sentiment classification in low-resource languages, with future improvements planned through data augmentation and transfer learning. Our full experimental codebase is publicly available at: github.com/ciol-researchlab/NAACL25-Mystic-Tamil-Sentiment-Analysis.

1 Introduction

Sentiment analysis, or opinion mining, is a critical task in Natural Language Processing (NLP) that involves determining the sentiment or emotional tone expressed in a given text. It plays a vital role in analyzing customer feedback, public opinion, and social media discourse, influencing industries such as marketing, politics, and e-commerce (Sebastiani, 2002). Traditional sentiment analysis methods relied on lexicon-based or machine learning approaches, but recent advancements in deep learning and transformer-based architectures have significantly improved performance. Models like BERT and SBERT can capture complex contextual relationships, making sentiment classification more accurate (Devlin, 2018; Reimers, 2019). However, while these models excel in high-resource languages like English, their effectiveness in low-resource languages remains limited. Sentiment analysis in underrepresented languages, such as Tamil and Tulu, presents unique challenges, including morphological richness, dialectal variations, and a lack of high-quality annotated datasets.

Sentiment analysis in underrepresented languages like Tamil and Tulu faces challenges such as morphological complexity, dialectal variations, and data scarcity. Despite being spoken by millions, these languages lack computational tools and annotated datasets (Joshi et al., 2020). Traditional classifiers and word embeddings struggle in resource-constrained settings (Hedderich et al., 2020). Additionally, code-switching and syntactic variations complicate sentiment classification, as monolingual models fail to capture linguistic nuances (Hegde et al., 2023b). Addressing these challenges requires adaptable frameworks that integrate advanced NLP techniques while mitigating data limitations.

To tackle these challenges, this study proposes a hybrid framework that integrates fine-tuned SBERT embeddings with a custom Multi-Layer Perceptron (MLP) classifier. The SBERT model generates high-dimensional contextual embeddings, capturing intricate linguistic patterns in Tamil and Tulu text. These embeddings are then processed using an optimized MLP classifier incorporating dropout regularization and ReLU activation to enhance robustness and prevent overfitting (Hwang and Jeong, 2023). The proposed framework achieves validation F1-scores of 0.4218 for Tamil and 0.3935 for Tulu, demonstrating its effectiveness in sentiment classification for low-resource languages. This research contributes to the broader field of NLP by

Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 254–259

establishing a scalable and adaptable classification pipeline. Future work will explore data augmentation, transfer learning, and hyperparameter tuning to further optimize performance, ultimately fostering more inclusive AI solutions for underrepresented linguistic communities (Feng et al., 2021).

2 Problem Description

Problem Statement. The 7th task, Sentiment Analysis in Tamil and Tulu, in the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages (DravidianLangTech-2025) at NAACL 2025 (Durairaj et al., 2025; Chakravarthi et al., 2020; Hegde et al., 2022, 2023a; S. K. et al., 2024) focuses on text classification in low-resource settings. Tamil and Tulu, despite their rich linguistic heritage, face significant challenges due to the scarcity of labeled datasets and the complexity of their syntactic and morphological structures. Traditional sentiment classification models struggle in these languages, particularly in multilingual and code-mixed contexts, where words from different languages are interwoven. Key challenges in low-resource text classification include limited high-quality annotated datasets, complicating supervised learning, and issues like code-mixing and dialectal variations, which introduce inconsistencies in syntax, morphology, and orthography. Small dataset sizes increase overfitting risks, reducing generalization. Additionally, pre-trained embeddings for these languages are often unavailable or underdeveloped, limiting the effectiveness of transformer-based models. Overcoming these challenges requires innovative approaches that utilize contextual embeddings while addressing data scarcity issues.

Dataset. This study utilizes Tamil and Tulu sentiment analysis datasets, split into training, validation, and test sets (Durairaj et al., 2025; Chakravarthi et al., 2020; Hegde et al., 2022, 2023a; S. K. et al., 2024). The datasets contain labeled textual inputs across various sentiment categories, helping to train and evaluate classification models effectively. Tamil has 31,122 training, 3,843 validation, and 3,459 test samples, while Tulu consists of 13,308 training, 1,643 validation, and 1,479 test samples. The test sets lack sentiment labels, requiring model predictions for evaluation. These datasets capture diverse linguistic structures, including code-mixed text, posing challenges for sentiment classification. A summary of dataset statistics is presented in Table 1.

Table 1: Overview of Tamil and Tulu datasets.

Dataset	Language	Entries
Training Set	Tamil	31,122
Validation Set	Tamil	3,843
Test Set	Tamil	3,459
Training Set	Tulu	13,308
Validation Set	Tulu	1,643
Test Set	Tulu	1,479

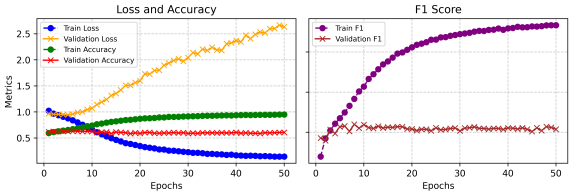

Figure 1: Training and validation metrics for Tamil language.

3 Methodology

We employed a comprehensive methodology to classify textual data effectively using an advanced hybrid pipeline that combines Sentence-BERT (SBERT) embeddings with a custom Multi-Layer Perceptron (MLP) architecture. Our approach leverages SBERT to generate rich, contextualized representations of Tamil and Tulu text, capturing semantic nuances often missed by traditional embeddings. These embeddings are then processed through an optimized MLP classifier, incorporating dropout regularization and ReLU activation to enhance robustness and prevent overfitting. This hybrid framework enables efficient sentiment classification in low-resource settings, addressing key challenges such as code-mixing and dialectal variations.

Data Preprocessing. We mapped textual sentiment labels to numerical values to ensure compatibility with machine learning models. For Tamil, we utilized the pre-trained SBERT model *l3cube-pune/indic-sentence-similarity-sbert* (Deode et al., 2023), while for Tulu, we employed *m3hrdadfi/zabanshenas-roberta-base-mix* (Farahani, 2021). These models are well-suited for capturing semantic similarities, making them effective for sentiment classification. Additionally, all preprocessing operations were conducted in a GPU-enabled environment to optimize efficiency and reduce computational overhead.

Embedding Generation. To generate embeddings,

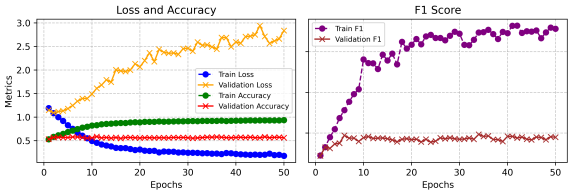

Figure 2: Training and validation metrics for Tulu language.

we tokenized the textual data using the specialized tokenizer from each respective SBERT model, ensuring proper padding and truncation to a maximum sequence length of 1024 tokens (Reimers, 2019). We extracted embeddings from the [CLS] token in the last hidden layer, as it provides a high-dimensional semantic representation of the input text (Devlin, 2018). These embeddings were then stored as .pt files to avoid redundant computations. For training and validation, the embeddings were paired with numerical labels, while test set embeddings were stored separately for later evaluation.

MLP Model Design. We designed two MLP models—one for Tamil and one for Tulu—to classify the high-dimensional embeddings effectively. Each MLP consisted of three fully connected layers with dimensions of 1024, 512, and the number of output classes, respectively. To enhance non-linearity, we applied ReLU activation functions after each hidden layer. Additionally, dropout layers with a probability of 0.3 were incorporated to mitigate overfitting (Srivastava et al., 2014). The final classification layer used a softmax activation function to generate class probabilities. For optimization, we utilized the Adam optimizer with a learning rate of 0.001 and CrossEntropyLoss as the loss function, ensuring stable convergence during training.

Training Process. We trained our model for 50 epochs, monitoring performance on the validation dataset after each epoch. To optimize memory usage and improve generalization, we shuffled the data and processed it in batches of 32. The model adjusted its weights using backpropagation, minimizing the loss function. We evaluated performance using accuracy, precision, recall, and F1-score (Powers, 2020), ensuring a comprehensive assessment of generalization capability. The model with the highest F1 score on the validation set was saved to disk, guaranteeing that only the most optimal version was used for final predictions. To maintain consistency across runs, we applied a fixed random seed.

Testing and Submission. For the test dataset, predictions were generated using the saved best model. The test embeddings were passed through the trained MLP, and class probabilities were computed. The final predictions were determined by selecting the class with the highest probability. A submission file containing the original text and its predicted label was prepared and saved in CSV format. This methodology ensured a systematic approach to embedding generation, model training, and evaluation, leveraging SBERT's semantic power and MLP's flexibility to achieve robust classification results (Deode et al., 2023; Farahani, 2021; Devlin, 2018).

4 Results and Discussion

Tamil Language Results. For the Tamil test dataset, predictions were generated using the best-performing model. The test embeddings were processed through the trained MLP, and class probabilities were computed. The final predictions were determined by selecting the class with the highest probability. A structured submission file was created, containing the original text and its predicted sentiment label, ensuring systematic evaluation. This methodology leveraged SBERT's semantic representation capabilities and MLP's adaptability to achieve robust classification results (Deode et al., 2023; Farahani, 2021; Devlin, 2018). During training, the Tamil language model exhibited consistent improvements in classification performance. By epoch 30, the training F1-score reached 0.8890, while the validation F1-score stabilized at 0.4218. Training beyond epoch 40 resulted in diminishing returns, with validation performance plateauing, suggesting that the model had effectively captured core linguistic patterns in the dataset. Precision and recall metrics showed balanced growth, with precision steadily increasing, indicating improved classification accuracy. However, recall demonstrated slight declines in later epochs, reflecting the challenge of maintaining generalization as the model became more confident in its predictions. Beyond epoch 40, the gap between training and validation performance widened, indicating overfitting tendencies (Ying, 2019). The training accuracy reached 93.22% by epoch 50, whereas validation accuracy remained at 56.42%, with a final validation F1-score of 0.3704. The validation loss peaked at 2.7500 by epoch 44, further confirming signs of overfitting.

Tulu Language Results. The Tulu language model followed a similar trend to the Tamil model, showing early improvements before reaching a plateau. Initial training began with a modest accuracy of 52.89% and an F1-score of 0.2897 at epoch 1. At this stage, validation accuracy and F1-score were recorded as 53.26% and 0.2879, respectively. By epoch 6, the model exhibited significant progress, achieving a training accuracy of 71.45% and an F1-score of 0.5126, while validation accuracy and F1-score reached 57.88% and 0.3887. The model's highest recorded validation F1-score of 0.3935 was observed at epoch 34 (Figure 2). However, beyond epoch 34, the improvements in both training and validation metrics became marginal, indicating saturation in learning. The final epoch (epoch 50) saw a training F1-score of 0.9004, but the validation F1-score plateaued at 0.3704, with the validation loss peaking at 2.7500 by epoch 44, pointing to signs of overfitting. To enhance the model's generalization, further optimization is needed. Implementing regularization techniques, such as dropout layers and L2 weight decay, can help reduce overfitting. Additionally, data augmentation strategies, such as generating synthetic text samples, could expand the dataset and improve model robustness. Leveraging transfer learning by utilizing multilingual pre-trained models is another promising approach to improving feature extraction and enhancing model performance (Ruder et al., 2019). These refinements aim to improve classification accuracy and ensure better generalization for unseen Tulu text, addressing the limitations observed during training.

Test Scores. We achieved test F1-scores of 0.4299 for Tamil and 0.1546 for Tulu. These results demonstrate the model's varying effectiveness across different languages.

5 Discussion

5.1 Performance Disparity Between Tamil and Tulu

The large gap in F1-scores between Tamil (0.4299) and Tulu (0.1546) stems from differences in resource availability. Tamil benefits from a well-pretrained model (`l3cube-pune/indic-sentence-similarity-sbert`) and a larger corpus, allowing better semantic representation. Tulu, in contrast, suffers from data scarcity and lacks a dedicated pre-trained model, leading to poor generalization and lower classification performance.

5.2 Overfitting Issues

The significant gap between training and validation performance, particularly in Tulu, indicates overfitting. Due to limited training data, the model memorizes patterns rather than learning generalizable features. The high model complexity relative to the dataset size, along with weak regularization, exacerbates this issue. Future improvements should include data augmentation, stronger regularization, and transfer learning to enhance generalization.

5.3 Linguistic Challenges

Tulu faces several linguistic challenges, including its low digital presence, morphological complexity, and lack of a standardized script. The frequent use of code-mixing with Kannada further complicates text classification. Unlike Tamil, which has well-structured textual data, Tulu's inconsistencies make NLP tasks more difficult. Addressing these challenges requires better datasets, improved tokenization, and language-specific embeddings.

5.4 Future Directions

Future work should focus on transfer learning with related languages, data augmentation for enhanced training diversity, and improved regularization to reduce overfitting. Most importantly, developing Tulu-specific models will be crucial for advancing NLP research in low-resource languages, ensuring greater inclusivity in AI applications.

6 Conclusion

This research proposed a hybrid pipeline for text classification in Tamil and Tulu, combining SBERT embeddings with custom MLP architectures. Despite limited resources, our methodology achieved promising results, with validation F1-scores of 0.4218 for Tamil and 0.3935 for Tulu, demonstrating the framework's ability to capture semantic patterns. While overfitting and metric saturation posed challenges, the model's early-stage improvements and balanced performance underscore its potential. Future work, including data augmentation and transfer learning, can further enhance accuracy and generalization. This study lays a solid foundation for advancing text classification in multilingual and low-resource settings, providing valuable insights for developing robust models in similar linguistic environments.

Limitations

One limitation of this work is the reliance on small, potentially non-representative datasets, which can hinder the model's ability to generalize effectively across diverse real-world scenarios. While the model performed well in the initial stages, issues like overfitting, particularly in the Tulu language model, highlight the challenge of training deep learning models on limited data. Moreover, the absence of high-quality annotated resources for these languages remains a significant barrier to achieving optimal performance.

Broader Impact Statement

Despite these challenges, the broader impact of this research is substantial. By developing a hybrid pipeline for text classification in low-resource languages like Tamil and Tulu, the work lays a foundation for more inclusive AI applications. Enhancing language accessibility through AI can bridge gaps in sentiment analysis, content moderation, and social media monitoring, especially for regional languages. This research demonstrates the potential to create more equitable and culturally aware AI systems that can serve underrepresented linguistic communities worldwide.

Acknowledgement

We express our sincere gratitude to Computational Intelligence and Operations Laboratory (CIOL) for their invaluable guidance, unwavering support, and continuous assistance throughout this journey. We are deeply appreciative of their efforts in organizing the CIOL Winter ML Bootcamp (Wasi et al., 2024), which provided an enriching learning environment and a strong foundation for collaborative research. The research mentoring and structured support offered by CIOL played a pivotal role in shaping this work, fostering innovation, and empowering participants to contribute meaningfully to the field of computational linguistics.

References

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John Philip McCrae. 2020. Corpus creation for sentiment analysis in code-mixed Tamil-English text. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 202–210, Marseille, France. European Language Resources association.

Samruddhi Deode, Janhavi Gadre, Aditi Kajale, Ananya Joshi, and Raviraj Joshi. 2023. L3cube-indicsbert: A simple approach for learning cross-lingual sentence representations using multilingual bert. *arXiv preprint arXiv:2304.11434*.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Thenmozhi Durairaj, Bharathi Raja Chakravarthi, Asha Hedge, Hosahalli Lakshmaiah Shashirekha, Rajeswari Natarajan, Sajeetha Thavareesan, Ratnasingam Sakuntharaj, Kalyanasundaram Krishnakumari, Charumathi Rajkumar, Poorvi Shetty, and Harshitha S Kumar. 2025. Overview of the Shared Task on Sentiment Analysis in Tamil and Tulu: DravidianLangTech@NAACL 2025. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Mehrdad Farahani. 2021. Zabanshenas is a solution for identifying the most likely language of a piece of written text.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*.

Michael A Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2020. A survey on recent approaches for natural language processing in low-resource scenarios. *arXiv preprint arXiv:2010.12309*.

Asha Hegde, Mudoor Devadas Anusha, Sharal Coelho, Hosahalli Lakshmaiah Shashirekha, and Bharathi Raja Chakravarthi. 2022. Corpus creation for sentiment analysis in code-mixed Tulu text. In *Proceedings of the 1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages*, pages 33–40, Marseille, France. European Language Resources Association.

Asha Hegde, Bharathi Raja Chakravarthi, Hosahalli Lakshmaiah Shashirekha, Rahul Ponnusamy, SUBALALITHA CN, Lavanya S K, Thenmozhi D, Martha Karunakar, Shreya Shreeram, and Sarah Aymen. 2023a. Findings of the shared task on sentiment analysis in tamil and tulu code-mixed text. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*, Varna, Bulgaria. Recent Advances in Natural Language Processing.

Asha Hegde, G Kavya, Sharal Coelho, Pooja Lamani, and Hosahalli Lakshmaiah Shashirekha. 2023b. Munlp@ dravidianlangtech2023: Learning approaches for sentiment analysis in code-mixed tamil and tulu text. In *Proceedings of the Third Workshop*

on Speech and Language Technologies for Dravidian Languages, pages 275–281.

Soon-Jae Hwang and Chang-Sung Jeong. 2023. Integrating pre-trained language model into neural machine translation. In *2023 2nd International Conference on Frontiers of Communications, Information System and Data Science (CISDS)*, pages 59–66. IEEE.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the nlp world. *arXiv preprint arXiv:2004.09095*.

David MW Powers. 2020. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Sebastian Ruder, Matthew E Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Tutorials*, pages 15–18.

Lavanya S. K., Asha Hegde, Bharathi Raja Chakravarthi, Hosahalli Lakshmaiah Shashirekha, Rajeswari Natarajan, Sajeetha Thavareesan, Ratnasingam Sakuntharaj, Thenmozhi Durairaj, Prasanna Kumar Kumaresan, and Charmathi Rajkumar. 2024. Overview of Second Shared Task on Sentiment Analysis in Code-mixed Tamil and Tulu. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, Malta. European Chapter of the Association for Computational Linguistics.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Azmine Toushik Wasi, MD Shakiqul Islam, Sheikh Ayatur Rahman, and Md Manjurul Ahsan. 2024. Ciol presnts winter ml bootcamp. 6 December, 2024 to 6 February, 2025.

Xue Ying. 2019. An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, page 022022. IOP Publishing.