MNLP@DravidianLangTech 2025: Transformers vs. Traditional Machine Learning: Analyzing Sentiment in Tamil Social Media Posts

Abhay Vishwakamra
Department of CSE
MNNIT-Allahabad
Prayagraj, Uttar Pradesh, 211004
vishwakarmaabhay10@gmail.com

Abhinav Kumar
Department of CSE
MNNIT-Allahabad
Prayagraj, Uttar Pradesh, 211004
abhik@mnnit.ac.in

Abstract

Sentiment analysis in Natural Language Processing (NLP) aims to categorize opinions in text. In the political domain, understanding public sentiment is crucial for influencing policymaking. Social media platforms like X (Twitter) provide abundant sources of real-time political discourse. This study focuses on political multiclass sentiment analysis of Tamil comments from X, classifying sentiments into seven categories: substantiated, sarcastic, opinionated, positive, negative, neutral, and none of the above. A number of traditional machine learning such as Complement Naive Bayes, Voting Classifier (an ensemble of Decision Tree, SVM, Naive Bayes, K-Nearest Neighbors, and Logistic Regression) and deep learning models such as LSTM, deBERTa, and a hybrid approach combining deBERTa embeddings with an LSTM layer are implemented. The proposed ensemble-based voting classifier achieved best performance among all implemented models with an accuracy of 0.3750, precision of 0.3387, recall of 0.3250, and macro-F_1-score of 0.3227.

1 Introduction

Sentiment analysis, a key task in Natural Language Processing (NLP), involves categorizing opinions in text (Kumar et al., 2020; Mishra et al., 2021). In the political domain, understanding public sentiment is crucial for policymaking. Social media platforms like X (formerly Twitter) provide real-time political discourse, but analyzing sentiments, particularly in code-mixed languages, presents unique challenges (Kumar et al., 2021, 2023; Kumari and Kumar, 2021). Code-mixing (Bokamba, 1988), common in multilingual communities, involves switching between languages in a single text. In India, users often blend English with regional languages like Tamil, creating challenges for sentiment analysis. Tamil, with its rich literary heritage, is frequently written in Roman script on social media, resulting in code-mixed content that complicates NLP tasks.

(Thavareesan and Mahesan, 2021) applied K-Means and KNN for sentiment analysis in Tamil texts. (Shanmugavadivel et al., 2022) evaluated machine learning models for sentiment classification in Tamil code-mixed tweets. (Anbukkarasi and Varadhaganapathy, 2020) explored deep neural networks, particularly DBLSTM, highlighting their ability to capture complex linguistic patterns.

A shared task on political multiclass sentiment analysis of Tamil social media posts was introduced during the DravidianLangTech@NAACL 2025 workshop (Durairaj et al., 2025). This task involved categorizing sentiments into seven categories: opinionated, sarcastic, substantiated, positive, negative, neutral, and none of the above. There are several deep learning models like LSTM, deBERTa, and a hybrid approach that combines deBERTa embeddings with an LSTM layer, as well as several traditional machine learning models like Complement Naive Bayes, Voting Classifier (an ensemble of Decision Tree, SVM, Naive Bayes, K-Nearest Neighbours, and Logistic Regression), and others are implemented.

The rest of the paper is summarized as follows: Section 2 introduces the dataset, Section 3 explains the proposed model, and the outcome of the proposed model is listed in Section 4, the discussion of findings and conclusion are listed in Section 5, limitations of proposed model and future directions are listed in Section 6.

2 Data Description

The dataset used for this analysis is provided by the DravidianLangTech@NAACL 2025 shared task[1], which focuses on political sentiment analysis in

[1] https://codalab.lisn.upsaclay.fr/competitions/20702#learn_the_details-overview

Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 482–486
May 3, 2025 ©2025 Association for Computational Linguistics

Tamil. It consists of a collection of Tamil-language tweets gathered from X, capturing a broad spectrum of political discussions. Each tweet is annotated with one of the seven sentiment categories: (i) substantiated, (ii) sarcastic, (iii) opinionated, (iv) positive, (v) negative, (vi) neutral, and (vii) none of the above. The overall distribution of the dataset is listed in Table 1.

Table 1: Distribution of Labels in Training and Validation Sets

Label	**Training**	**Validation**
Opinionated	1,361	153
Sarcastic	790	115
Neutral	637	84
Positive	575	69
Substantiated	412	52
Negative	406	51
None of the above	171	20
Total	**3,352**	**544**

3 Methodology

To tackle class imbalance issue, we used Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al., 2002) on our train dataset. Specifically, this method essentially generate artificial samples from linear combinations of two or more minority classes examples so that we can again have a more balanced sample.

Five different models were developed to identify hate or offensive contents in Dravidian posts; (i) Complement Naive Bayes, (ii) Voting Classifier, (iii) Long-Short Term-Memory (LSTM) , (iv) Transfer learning-based model, and (v) Hybrid model. In this section, we explain the working of each model in detail.

3.1 Complement Naive Bayes

We use the Complement Naive Bayes (Seref and Bostanci, 2019) (CNB) algorithm for text classification, ideal for imbalanced datasets as it adjusts weights using the complement of each class to reduce sensitivity to imbalances. Text data is preprocessed using Count Vectorizer, which converts text into a matrix of token counts representing word frequencies, serving as input for the classifier.

3.2 Voting Classifier

We use a Voting (Kuncheva and Rodríguez, 2014) Classifier with Decision Tree (Song and Ying, 2015), SVM (Jakkula, 2006), Multinomial Naive Bayes (Kibriya et al., 2005), K-Nearest Neighbors (Guo et al., 2003), and Logistic Regression (DeMaris, 1995). Soft voting (Cao et al., 2015) combines the predicted probabilities from each model for a balanced consensus. Text data is preprocessed using Count Vectorizer, which converts text into a matrix of token counts for training the ensemble.

3.3 Deep Learning Model

We built an LSTM (Aston Zhang, 2020) based model for text classification with two stacked (Wang et al., 2018) LSTM layers (64 and 32 units), dropout layers for regularization, and a softmax output for 7-class prediction. Text preprocessing included IndicNLP (Kunchukuttan) for tokenization and normalization, followed by padding for uniform input length.

3.4 Transfer Learning Model

We utilized a transfer learning approach with the DeBERTa (He et al., 2020) V3 base multilingual model for text classification into 7 categories. We use a DeBERTa model since its disentangled attention mechanism and absolute position embeddings yield better contextual embeddings and superior performance than traditional BERT-based models with common pooling methods especially in low-resource multilingual settings. The preprocessor was configured using *DebertaV3TextClassifierPreprocessor* with a sequence length of 64 and waterfall truncation then preprocessed input was fed into the *DebertaV3TextClassifier*, utilizing pre-trained embeddings for efficient predictions.

3.5 Hybrid Model

We use a hybrid model that combines DeBERTa V3 with LSTM (Rai et al., 2022) to utilize their complementary strengths. DeBERTa generates contextual embeddings, while LSTM captures sequential dependencies, enhancing performance for text classification tasks.

The input text is preprocessed using the DeBERTa preprocessor, which tokenizes and prepares data for the transformer. The DeBERTa V3 model outputs embeddings with a shape of batch_size, sequence_length, embedding_dim i.e.(32, 64, 768), which are fed into an LSTM layer. The first LSTM layer, with 128 units and second LSTM layer with 64 units. Finally, a dense classification layer with 7

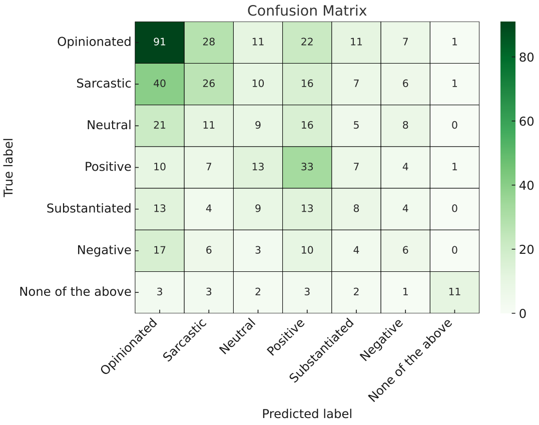

Figure 1: Confusion Matrix for Naive Bayes classifier

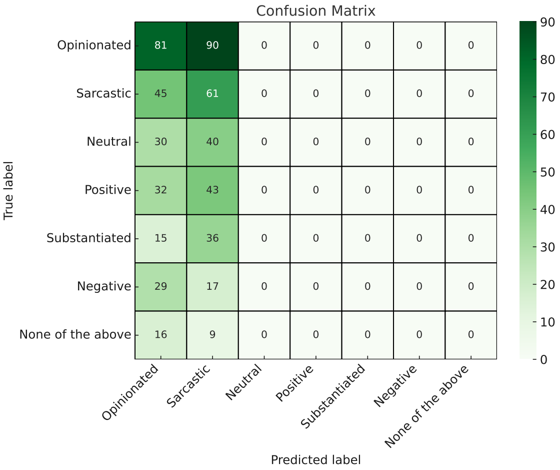

Figure 2: Confusion Matrix for DeBERTa model

units applies a softmax activation to produce class probabilities for predictions.

4 Result

This section has the results of the five models evaluated using accuracy, precision, recall, and macro-F_1-scores (Opitz and Burst, 2019). The results of different machine learning and deep learning models can be seen in Table 2. As can be seen in Table 2, the Voting Classifier outperforms the other models with an accuracy of 0.3750, precision of 0.3387, recall of 0.3250 and an macro-F_1-score of 0.3227. The Naive Bayes classifier shown good efficiency for text classification with an accuracy of 0.3382, precision of 0.3367, recall of 0.2962, and macro-F_1-score of 0.3059. Similar results were obtained by the hybrid model (DeBERTa + LSTM), which benefited from both contextual and sequential learning, with an accuracy of 0.3162, precision of 0.3143, recall of 0.2997, and macro-F_1-score of 0.3026.

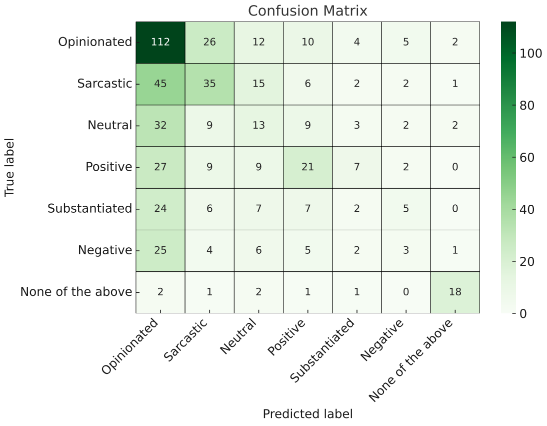

Figure 3: Confusion Matrix for Voting Classifier

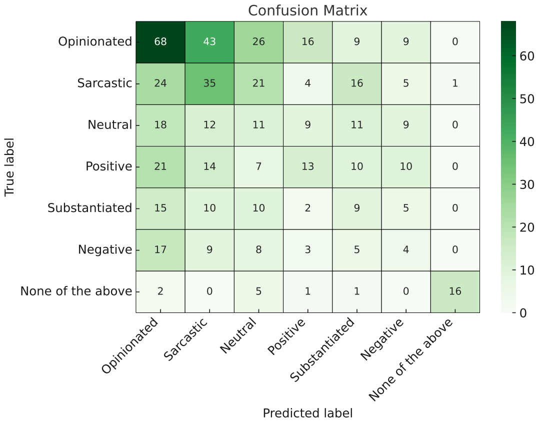

Figure 4: Confusion Matrix for LSTM model

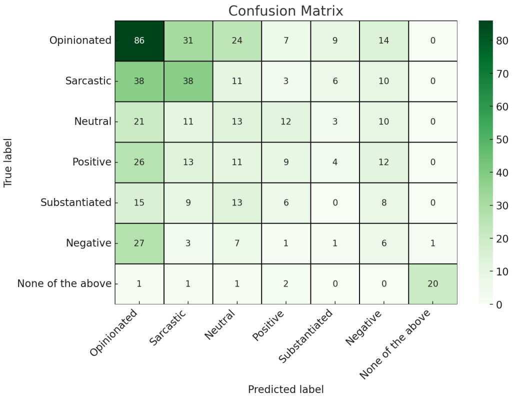

Figure 5: Confusion Matrix for Hybrid Model (DeBERTa with LSTM).

Table 2: Performance comparison of different models.

Model	Accuracy	Precision	Recall	macro-F_1-score
Naive Bayes	0.3382	0.3367	0.2962	0.3059
Voting Classifier	0.3750	0.3387	0.3250	0.3227
2-layer LSTM	0.2868	0.3252	0.2803	0.2964
DeBERTa_v3	0.2610	0.0761	0.1499	0.0986
DeBERTa Embeddings + LSTM layer	0.3162	0.3143	0.2997	0.3026

The 2-layer LSTM model showed moderate results, with an accuracy of 0.2868, precision of 0.3252, recall of 0.2803, and macro-F_1-score of 0.2964. The DeBERTa v3 model had the lowest metrics, with an accuracy of 0.2610, precision of 0.0761, recall of 0.1499, and macro-F_1-score of 0.0986, indicating that pre-trained embeddings alone might not fully capture the task's nuances. Overall, ensemble and hybrid models, like the Voting Classifier and DeBERTa embedding with LSTM, performed better compared with other implemented models.

5 Discussion and Conclusion

The confusion matrix (Heydarian et al., 2022) for Naive Bayes shows some misclassification (see Figure 1), particularly for Opinionated, Sarcastic, and Neutral classes. The Transfer Learning Model (DeBERTa) exhibits a high degree of misclassification (see Figure 2), especially for classes Opinionated, Sarcastic, and Neutral. The confusion matrix (see Figure 3) for the Voting Classifier shows a good balance of correct predictions and minimal misclassifications. The Deep Learning (2 stacked LSTM Layers) (see Figure 4) and Hybrid (DeBERTa embedding with LSTM) (see Figure 5) models show moderate performance with some misclassifications. The Hybrid model appears to have slightly better performance than the LSTM based on the confusion matrix.

The Voting Classifier performs best with the highest accuracy, precision, and recall. Naive Bayes and deBERTa show weaker results, while the LSTM and Hybrid models perform moderately. Overall, ensemble methods like the Voting Classifier are most effective for political sentiment analysis in Tamil tweets.

6 Limitations and Future Work

Since the oversampling of SMOTE might affect a model, class imbalance, continues to remain a challenge, as SMOTE generates synthetic samples alike to the real-world scenario but does not necessarily denote the complexities of natural language. As a result, the dataset is large and domain-specific, and providing additional annotated samples for dominant classes in particular can enhance the robustness of the dataset. Finally, real-time deployment of DeBERTa incurs considerable computational costs given its transformer architecture, so future work may explore lower-cost architectures or distillation (Gou et al., 2021) approaches.

In order to overcome these limitations, we intend to extend the dataset, investigate multimodal approaches in addition to text, further fine-tune model efficiency, and evaluate domain transfer over various Tamil dialects along with variations in political discourse.

The code for the proposed framework is available at:
https://github.com/abhay-43/Deep-Learning-Approach-for-Analyzing-Sentiment-in-Tamil-Social-Media-Posts.git

References

S Anbukkarasi and S Varadhaganapathy. 2020. Analyzing sentiment in tamil tweets using deep neural network. In *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, pages 449–453. IEEE.

Mu Li Alexander J. Smola Aston Zhang, Zachary C. Lipton. 2020. Dive into deep learning. `https://d2l.ai/chapter_recurrent-modern/lstm.html`. Accessed: 2024-11-26.

Eyamba G Bokamba. 1988. Code-mixing, language variation, and linguistic theory:: Evidence from bantu languages. *Lingua*, 76(1):21–62.

Jingjing Cao, Sam Kwong, Ran Wang, Xiaodong Li, Ke Li, and Xiangfei Kong. 2015. Class-specific soft voting based multiple extreme learning machines ensemble. *Neurocomputing*, 149:275–284.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Alfred DeMaris. 1995. A tutorial in logistic regression. *Journal of Marriage and the Family*, pages 956–968.

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.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. 2003. Knn model-based approach in classification. In *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*, pages 986–996. Springer.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Mohammadreza Heydarian, Thomas E Doyle, and Reza Samavi. 2022. Mlcm: Multi-label confusion matrix. *IEEE Access*, 10:19083–19095.

Vikramaditya Jakkula. 2006. Tutorial on support vector machine (svm). *School of EECS, Washington State University*, 37(2.5):3.

Ashraf M Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. 2005. Multinomial naive bayes for text categorization revisited. In *AI 2004: Advances in Artificial Intelligence: 17th Australian Joint Conference on Artificial Intelligence, Cairns, Australia, December 4-6, 2004. Proceedings 17*, pages 488–499. Springer.

Abhinav Kumar, Sunil Saumya, and Jyoti Prakash Singh. 2020. NITP-AI-NLP@ Dravidian-CodeMix-FIRE2020: a hybrid cnn and bi-lstm network for sentiment analysis of Dravidian code-mixed social media posts. In *FIRE (Working Notes)*, pages 582–590.

Abhinav Kumar, Sunil Saumya, and Jyoti Prakash Singh. 2021. An ensemble-based model for sentiment analysis of Dravidian Code-Mixed social media posts. In *FIRE (Working Notes)*, pages 950–958.

Abhinav Kumar, Jyoti Prakash Singh, and Amit Kumar Singh. 2023. Explainable BERT-LSTM stacking for sentiment analysis of covid-19 vaccination. *IEEE Transactions on Computational Social Systems*.

Jyoti Kumari and Abhinav Kumar. 2021. A deep neural network-based model for the sentiment analysis of dravidian code-mixed social media posts. *management*, 5:6.

Ludmila I Kuncheva and Juan J Rodríguez. 2014. A weighted voting framework for classifiers ensembles. *Knowledge and information systems*, 38:259–275.

Anoop Kunchukuttan. Indic nlp library. https://anoopkunchukuttan.github.io/indic_nlp_library/.

Ankit Kumar Mishra, Sunil Saumya, and Abhinav Kumar. 2021. Sentiment analysis of Dravidian-CodeMix language. In *FIRE (Working Notes)*, pages 1011–1019.

Juri Opitz and Sebastian Burst. 2019. Macro f1 and macro f1. *arXiv preprint arXiv:1911.03347*.

Nishant Rai, Deepika Kumar, Naman Kaushik, Chandan Raj, and Ahad Ali. 2022. Fake news classification using transformer based enhanced lstm and bert. *International Journal of Cognitive Computing in Engineering*, 3:98–105.

Berna Seref and Erkan Bostanci. 2019. Performance comparison of naïve bayes and complement naïve bayes algorithms. In *2019 6th international conference on electrical and electronics engineering (ICEEE)*, pages 131–138. IEEE.

Kogilavani Shanmugavadivel, Sai Haritha Sampath, Pramod Nandhakumar, Prasath Mahalingam, Malliga Subramanian, Prasanna Kumar Kumaresan, and Ruba Priyadharshini. 2022. An analysis of machine learning models for sentiment analysis of tamil code-mixed data. *Computer Speech & Language*, 76:101407.

Yan-Yan Song and LU Ying. 2015. Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 27(2):130.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2021. Sentiment analysis in tamil texts using k-means and k-nearest neighbour. In *2021 10th International Conference on Information and Automation for Sustainability (ICIAfS)*, pages 48–53. IEEE.

Jin Wang, Bo Peng, and Xuejie Zhang. 2018. Using a stacked residual lstm model for sentiment intensity prediction. *Neurocomputing*, 322:93–101.