Wictory@DravidianLangTech 2025: Political Sentiment Analysis of Tamil X(Twitter) Comments using LaBSE and SVM

Nithish Ariyha K, Eshwanth Karti T R,Yeshwanth Balaji AP , Vikash J, Sachin Kumar S
Amrita School of Artificial Intelligence, Coimbatore
Amrita Vishwa Vidyapeetham, India
cb.en.u4aie22140@cb.students.amrita.edu,
cb.en.u4aie22118@cb.students.amrita.edu,
cb.en.u4aie22102@cb.students.amrita.edu,
cb.en.u4aie22156@cb.students.amrita.edu, s_sachinkumar@cb.amrita.edu

Abstract

Political sentiment analysis has become an essential area of research in Natural Language Processing (NLP), driven by the rapid rise of social media as a key platform for political discourse. This study focuses on sentiment classification in Tamil political tweets, addressing the linguistic and cultural complexities inherent in low-resource languages. To overcome data scarcity challenges, we develop a system that integrates embeddings with advanced Machine Learning techniques, ensuring effective sentiment categorization. Our approach leverages deep learning-based models and transformer architectures to capture nuanced expressions, contributing to improved sentiment classification. This work enhances NLP methodologies for low-resource languages and provides valuable insights into Tamil political discussions, aiding policymakers and researchers in understanding public sentiment more accurately. Notably, our system secured **Rank 5** in the NAACL shared task, demonstrating its effectiveness in real-world sentiment classification challenges.

Keywords: political sentiment, NLP, SVM, LaBSE, MuRIL, transformer, deep learning, sentiment classification

1 Introduction

The digital age has ushered into society social media, which dramatically changed the discourse of political issues and ushered in totally unprecedented avenues to involve the public within them. Social networking sites, especially platform forums like X are dynamic debating forums that offer a varied mix of thought streams to join a continuous flowing debate. In the digital age, regional languages like Tamil have become indispensable archives of grass-root-level political discourse that unfold genuine insights into local viewpoints and culture-related politically complex expressions. (Gioia, 2023)

A recent area in NLP emerged under the rubric of political sentiment analysis. This aims at providing much-needed insights into the public sentiment by methodical categorization of textual representations of political opinion. Since it supports the measurement of the public's response to policies, measures political engagement, and identifies critical issues in society, this kind of analysis has a high utility for policymakers, political analysts, and governmental agencies. With sentiment analysis, proper policy solutions would be devised more effectively and with alignment to community requirements.

But sentiment classification, per se, is inherently challenging because it often muddles multiple tones: sarcasm, strong opinions, or even seemingly neutral observations. In low-resource languages like Tamil, these challenges are compounded by linguistic and cultural complexities demanding sophisticated techniques for capturing nuances in expression.

This work hopes to address such issues by partitioning Tamil political tweets into seven different groups that support data analysis. In doing so, we look forward to contributing to the further development of NLP methods for low-resource languages while increasing our understanding of Tamil political mood at the same time. This review also aims to provide a comparative analysis of different models and techniques for this task. This work is based on the shared task in DravidanLangTech2025@NAACL (Chakravarthi et al., 2025).

2 Related Works

Political sentiment analysis has gained attention with the rise of social media, particularly X. Review by (Wankhade et al., 2022) discusses about senti-

Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 741–745

ment analysis in different areas including social media and e-commerce, methodologies, applications, and challenges. It emphasizes methods like lexicon-based, ML, and hybrid approaches while addressing issues like sarcasm, ambiguity, and language-specific challenges. The study also highlights the impact of emoticons and emojis, which prompted the use of embeddings with ML models for sentiment analysis.

(Elghazaly et al., 2016) compared the Naïve Bayes classifier and SVM classifiers on Arabic tweets in the 2012 Egyptian elections, solving problems like inflectional variation, stemming, and sarcasm. (Babu, 2022) experimented with Tamil sentiment classification on movie reviews using CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU and obtained the maximum accuracy with CNN-BiLSTM. Kumar S (Kumar S et al., 2017) employed CNN and LSTM to classify Malayalam tweets and got better results for identification tasks.

(Tripty et al., 2024) explored ML and DL models for sentiment analysis of YouTube comments, highlighting the strong performance of encoder models like XLM-RoBERTa and IndicBERT. (Kannan et al., 2021) applied IndicBERT to code-mixed Tamil tweets, achieving a 61.73 F1-score, which aligns with our task of classifying political sentiments in Tanglish data.

(Tripty et al., 2024) explored a variety of ML and DL based models for sentiment analysis of youtube comments. Their review revealed how encoder models like XLM-RoBERTa and IndicBERT perform well in the classification task, paving way for model selection in our work. The authors of (Kannan et al., 2021) apply Indic-BERT to analyze code-mixed Tamil tweets and demonstrates its effectiveness over traditional methods. The study achieved an F1 score of 61.73, and this aligns closely with the task of classification of political sentiments for Tanglish data (Tamil and English).

(Kumar and Albuquerque, 2021)'s study shows the performance of XLM-R large model in comparison with models like BB_Twtr and DataStories . The XLM-R large model surpasses the rest models by 5% with 71.8% accuracy. Authors of (Nithya et al., 2022) aim to apply deep learning based Bi-LSTM model with ULMFiT for sentiment analysis, which gave them promising and better results. Authors of (Shanmugavadivel et al., 2022) provide and analysis of multiple machine learning models for sentiment analysis of Tamil code-mixed data. They tested many methods like SVM, Logistic Regressions, CNN, BiLSTM and many with their best F1 score being around 0.66.

3 Dataset

The dataset used in this study was obtained from the 2025 Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics. The dataset is pre-divided into training and test subsets, with the test set comprising approximately 430 sentences. The training dataset consists of about 4,300 Tamil sentences, where each sentence contains both hashtags and emojis, categorized into seven distinct classes: Substantiated, Sarcastic, Opinionated, Positive, Negative, Neutral, and None of the Above. The lengths of the sentences are predominantly between 100 and 250 characters. With the dataset size being small, it is also imbalanced. It contains more samples in Opinionated than in the None of the Above category.

4 Methodology

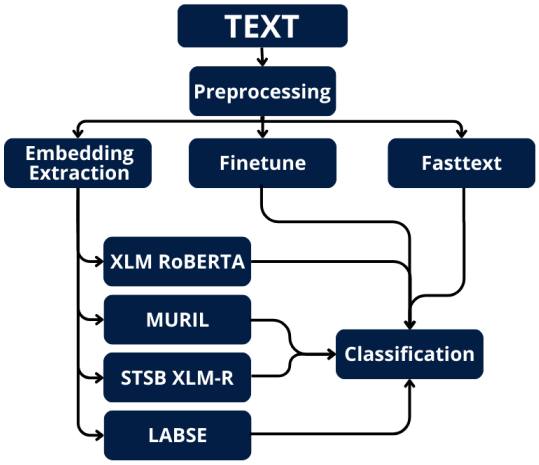

Figure 1: Experimentation pipeline

Fig. 1 shows the experimentation pipeline that was followed.

4.1 Data Preprocessing

The preprocessing techniques of this study center on cleaning and normalizing textual data for better quality and reliability of downstream NLP tasks. First, emojis are replaced by their text equivalent to ensure contextual information. Hashtags are replaced with the tag <*hashtag*> for maintaining tweet data. As the emojis contain meaningful sentiment-related information, simply deleting or replacing with a generic tag as done with hashtags

would compromise model performance. All text is further converted to lowercase, removing special characters, extra spaces (including tabs and newlines), URLs, and mentions to improve uniformity. Other words with certain symbols like currency signs or special characters are also removed, as they could interfere in unwanted ways during text analysis. Overall, all these preprocessing steps lead to noise reduction, text normalization, and an overall improvement in the quality of data. Finally, labels and content are formatted according to requirements, such as converting labels to numerical values or adapting them to FastText format.

To improve the class imbalance we tried the method of ADASYN(Adaptive Synthetic Sampling). ADASYN is an oversampling technique that generates synthetic samples for the minority class based on data distribution (He et al., 2008). This method was used only in one trial due to the constraints of time and compute available.

4.2 Fine-Tuning Transformer models

Transformer-based models such as IndicBERT, MuRIL, TamilSBERT (Joshi et al., 2022), XLM-RoBERTa, and TamilBERT4MLM were trained with Adam optimizer and weight decay, batch size 8, and sequence length of 256(chose based on text distribution). Focal Loss was used to address class imbalance, while exploding gradients were avoided with gradient clipping . FastText's skip-gram model employed 300-dimensional vectors and a subword n-gram of three to six characters to adapt to the morphological richness of Tamil and code-mixed tokens (Bojanowski et al., 2017). F1-score was the primary evaluation metric.

4.3 Embedding with SVM Classifier

To effectively classify text using Support Vector Machines (SVM), we first extracted meaningful embeddings from multiple transformer-based and static embedding models. The embeddings served as input features to the SVM classifier, ensuring that the model had a well-represented feature space for learning. Below, we provide details on each stage of this process.

4.3.1 Embedding Extraction

To generate high-quality embeddings from each model, we employed distinct extraction strategies tailored to their respective architectures.

XLM-RoBERTa: Mean pooling over the last hidden state to obtain sentence representations that capture linguistic structures effectively. (Conneau et al., 2019)

LaBSE: Sentence representations were extracted and normalized to ensure uniform magnitude across different inputs, enhancing stability in classification. (Feng et al., 2022)

MuRIL: The classification token $[CLS]$ representation was used, leveraging its pre-training to encode sentence-level semantics efficiently.

STSb-XLM-R: Token embeddings were mean-pooled to generate comprehensive sequence representations.

4.3.2 ML Based Classification

For classification, we employed a Support Vector Machine (SVM) approach, using Scikit-learn's LinearSVC implementation to ensure computational efficiency and reliable performance.

To handle class imbalance, class weights were set inversely proportional to class frequencies. This strategy prevented minority classes from being underrepresented during training, ensuring that the classifier made balanced predictions across all categories.

Features were normalized employing Robust Scaler. Grid search tuned the regularization parameter $\{0.1, 1, 10\}$ to harmonize margin maximization and performance in classification. Combining structured embeddings with a classifier based on SVM efficiently made sentiment classification operational in Tamil-English code-mixed political utterances.

After the shared task closure, XGBoost was also experimented. It builds decision tree sequentially, correcting previous errors while minimizing loss. With L1/L2 regularization if handles overfitting. It is efficient in handling missing values, and parallel processingmakes it is highly scalable. LABSE embeddings trained the model using TF-IDF, enriching the input with more information.

5 Results and Observations

All transformer models had good accuracy on the initial stages of training, but the accuracy started to saturate at around 30%, indicating a trade-off limitation between overfitting and bad generalization beyond some point. With increasing overfitting to the training data, which is seen to grow large with respect to the gap between training and validation performance, the tested models produced different performances. TamilSBERT performed the best at an accuracy of 38% followed by F1 Score 0.26,

Table 1: Embedding Models and their Performance

Model	Weighted F1
FastText	0.280*
XLM-RoBERTa Base + SVM	0.220
Sentence-Transformers-LaBSE + SVM	**0.310***
MuRIL + SVM	0.150
Indic-Bert + SVM	0.24
STSb-XLM-R	0.260
Sentence-Transformers-LaBSE + XG-Boost + TFIDF + ADASYN	**0.330***

Weighted F1 Score

showing slight improvement over the others. The highest accuracy in TamilSBERT was due to a better tokenizer as it could store all word data while other tokenizers could not. The overfitting problem continued with all the models, showing that there is a need for better and diversified training data for better generalization of transformer-based architectures for Tamil English code-mixed political discourse.

From the outcome as shown in Table 1, the Sentence-Transformers-LaBSE model along with the SVM classifier performs best in SVM, reporting the highest value with an accuracy of 0.310 in the case of a weighted F1 score. This shows, LaBSE embeddings specifically optimized for sentence-level multilingual tasks are able to handle the text in Tamil English code-mixed variety, especially in low-resource settings. Post Shared task experiments showed XGBoost with LABSE and TF-IDF displayed the highest F1 score of 0.33

FastText performed well with a weighted F1 score of 0.280, showing its effectiveness in handling morphologically rich languages like Tamil. Although it is a multilingual model, XLM-RoBERTa Base scored a lower score at 0.220, possibly because it has not been exposed to much Tamil English code-mixed data. STSb-XLM-R showed moderate performance with an F1 score of 0.260, showing its ability to capture contextual relationships. Surprisingly, MuRIL, designed for Indian languages, had the lowest score (0.150), suggesting its pretraining may not sufficiently cover Tamil English code-mixed text. On looking close into the classification scores for each class a common pattern was observed. The class of none of the above showed a significantly high score of 0.79 whereas the class substantiated achieved only 0.11. The primary reason for this being the similarity in substantiated and opinionated. This could be the primary reason for average F1 scores.

The very low F1 values for Tamil-English code-mixed political opinion analysis can be attributed to a variety of reasons. The primary reason is the lack of sufficient and quality labeled data, which affects the ability of the model to learn informative patterns. A improper train-test split, with the test set including entirely unseen tokens, will also prevent generalization. The intricacy of code-mixed language, such as variations in grammar, inconsistency in transliteration, and varying word orders, makes it even more challenging. Most pretrained language models, also, are not specially trained for Tamil-English code-mixed data, which restricts their performance. A potential future improvement is using large language models to address these issues effectively.

Even though there are numerous research and reviews on sentiment analysis of Tamil-English code-mixed text, we couldn't compare our results with them due to their simple classification approach. The majority of work in this area focuses on basic sentiment analysis, such as classifying text as positive or negative, rather than a more detailed classification. This highlights a greater scope for future research in this area.

6 Conclusion

This report presents the findings from the sentiment analysis task conducted as part of the FifthWorkshop on Speech and Language Technologies .The task focused on classifying political sentiments in Tamil English code-mixed tweets, with the dataset provided by the conference. Our proposed method achieved a rank of 5th in the overall task. The results demonstrate the effectiveness of leveraging both transformer-based models and traditional embeddings for sentiment classification in low-resource languages, while highlighting the need for further improvements in handling code-mixed text for better generalization and better datasets.

Link for GitHub repository with codes[1]

[1] https://github.com/ariyha/NAACL-2025-Political-Sentiment-Analysis

7 Limitations

The volume of the given dataset could be expanded to capture greater variability, ensuring that deep learning models are trained on a more diverse representation of political discourse. Additionally, political tweets often include multimodal elements such as images, videos, memes, and emojis, which are not accounted for in text-only sentiment analysis models, potentially leading to incomplete or inaccurate sentiment predictions. Another crucial limitation is the issue of concept drift, where models trained on past data may become outdated as political narratives evolve over time. Therefore, sentiment models should not be static; they must be continuously updated to adapt to shifts in public opinion and emerging political contexts. This ongoing evolution is essential for real-world applications, where accurate sentiment analysis depends on the model's ability to reflect current sociopolitical dynamics rather than relying solely on historical data.

References

Suba Sri Ramesh Babu. 2022. Sentiment analysis in tamil language using hybrid deep learning approach. Msc research project, National College of Ireland.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Preprint*, arXiv:1607.04606.

Bharathi Raja Chakravarthi, Saranya Rajiakodi, Elizabeth Sherly, Thenmozhi Durairaj, Sathiyaraj Thangasamy, Ratnasingam Sakuntharaj, Prasanna Kumar Kumaresan, Kishore Kumar Ponnusamy, Arunaggiri Pandian Karunanidhi, and Rohan R. 2025. Overview of the Shared Task on Political Multiclass Sentiment Analysis of Tamil X(Twitter) Comments: DravidianLangTech@NAACL 2025. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Tarek Elghazaly, Amal Mahmoud, and Hesham A. Hefny. 2016. Political sentiment analysis using twitter data. In *Proceedings of the International Conference on Internet of Things and Cloud Computing*, ICC '16, New York, NY, USA. Association for Computing Machinery.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding. *Preprint*, arXiv:2007.01852.

Elio Simone La Gioia. 2023. Using sentiment analysis for politics: the case of the italian political elections.

Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. 2008. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328.

Ananya Joshi, Aditi Kajale, Janhavi Gadre, Samruddhi Deode, and Raviraj Joshi. 2022. L3cube-mahasbert and hindsbert: Sentence bert models and benchmarking bert sentence representations for hindi and marathi. *arXiv preprint arXiv:2211.11187*.

R. Ramesh Kannan, Ratnavel Rajalakshmi, and Lokesh Kumar. 2021. Indicbert based approach for sentiment analysis on code-mixed tamil tweets. In *Fire*.

Akshi Kumar and Victor Hugo C Albuquerque. 2021. Sentiment analysis using xlm-r transformer and zero-shot transfer learning on resource-poor indian language. *Transactions on Asian and Low-Resource Language Information Processing*, 20(5):1–13.

Sachin Kumar S, Anand Kumar Madasamy, and Soman Kp. 2017. *Sentiment Analysis of Tweets in Malayalam Using Long Short-Term Memory Units and Convolutional Neural Nets*, pages 320–334.

K. Nithya, S. Sathyapriya, M. Sulochana, S. Thaarini, and C. R. Dhivyaa. 2022. Deep learning based analysis on code-mixed tamil text for sentiment classification with pre-trained ulmfit. In *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 1112–1116.

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.

Zannatul Tripty, Md. Nafis, Antu Chowdhury, Jawad Hossain, Shawly Ahsan, Avishek Das, and Mohammed Moshiul Hoque. 2024. CUETSentimentSillies@DravidianLangTech-EACL2024: Transformer-based approach for sentiment analysis in Tamil and Tulu code-mixed texts. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 234–239, St. Julian's, Malta. Association for Computational Linguistics.

Mayur Wankhade, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. 2022. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, 55(7a):5731–5780.