AnalysisArchitects@DravidianLangTech 2025: Machine Learning Approach to Political Multiclass Sentiment Analysis of Tamil

Abirami Jayaraman **Aruna Devi Shanmugam** **Dharunika Sasikumar**
abirami2210382@ssn.edu.in aruna2210499@ssn.edu.in dharunika2210459@ssn.edu.in
Bharathi B
bharathib@ssn.edu.in

Department of Computer Science and Engineering
Sri Sivasubramaniya Nadar College of Engineering
Kalavakkam, Chennai, Tamil Nadu

Abstract

Sentiment analysis is recognized as an important area in Natural Language Processing (NLP) that aims at understanding and classifying opinions or emotions in text. In the political field, public sentiment is analyzed to gain insight into opinions, address issues, and shape better policies. Social media platforms like Twitter (now X) are widely used to express thoughts and have become a valuable source of real-time political discussions. In this paper, the shared task of Political Multiclass Sentiment Analysis of Tamil tweets is examined, where the objective is to classify tweets into specific sentiment categories. The proposed approach is explained, which involves preprocessing Tamil text, extracting useful features, and applying machine learning and deep learning models for classification. The effectiveness of the methods is demonstrated through experimental results and the challenges encountered while working on the analysis of Tamil political sentiment are discussed.

1 Introduction

Sentiment analysis has become a pivotal area within Natural Language Processing (NLP), focusing on identifying sentiments expressed in text. This capability is particularly valuable in political contexts, where understanding public sentiment can guide decision-making and policy improvements. Social media platforms, such as Twitter (now X), serve as rich sources of real-time data, offering diverse perspectives on political issues. Analyzing public discourse on these platforms provides essential information on the emotions and opinions of the population.

This study specifically focuses on the analysis of political multiclass sentiment in Tamil tweets. Unlike general sentiment analysis, this task requires the classification of tweets into multiple sentiment categories tailored to the political domain. The complexity of working with Tamil text is further compounded by challenges such as code-mixing, spelling variations, and a scarcity of annotated datasets. To address these challenges, various models, including machine learning and deep learning techniques, were tested. The Naïve Bayes algorithm was used for its simplicity and effectiveness in text classification, Support Vector Machines (SVM) were used for their robustness in handling high-dimensional data, and Long-Short-Term Memory (LSTM) networks were used to capture sequential patterns in text data.

Sentiment analysis was applied effectively to political contexts in various languages. For example, Gunhal's study (Gunhal, 2023a) on the sentiment surrounding Karnataka's elections used transformer-based models to analyze Twitter data, achieving significant precision in predicting electoral results. This is consistent with previous research demonstrating the predictive influence of Twitter sentiment on electoral results. Furthermore, previous studies by Krishnan(Krishnan et al.) have highlighted the importance of sentiment analysis in understanding public reactions to political events and policies.

The paper is structured as follows: Section 2 presents an overview of related works, summarizing existing research on sentiment analysis and Tamil text classification. Section 3 describes the dataset used, including its sources and preprocessing steps. In Section 4, we detail our methodology encompassing data preprocessing, feature extraction, and the models employed. Section 5 provides detailed description about the implementation of the work proposed. Section 6 presents our experimental findings and performance metrics used for evaluation. Section 7 elaborates the shortcomings of these models in general. Finally, Section 8 concludes with a summary of the work done and it's

Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 666–670

result in Tamil political sentiment analysis.

2 Related Works

The study by Pranav Gunhal(Gunhal, 2023b) investigates sentiment classification surrounding the 2023 Karnataka elections using transformer-based models, particularly IndicBERT. This research explores innovative data collection and augmentation techniques to analyze Twitter sentiment, classifying posts as positive, negative, or neutral. The findings demonstrate the potential of these models in forecasting electoral outcomes and capturing sentiment trends in Indian politics, highlighting their value for political stakeholders in future elections.

Rajasekar et al(Rajasekar and Geetha, 2023) present a robust deep learning model specifically designed for analyzing Tamil tweets. The authors aim to improve sentiment classification accuracy using deep convolutional neural networks (CNNs). The model significantly outperforms traditional methods, demonstrating its effectiveness in capturing sentiment nuances within Tamil language tweets.

Chakravarthi et al. (Chakravarthi et al., 2025) presented a machine learning-based approach for political multiclass sentiment analysis of Tamil X (Twitter) comments as part of the DravidianLangTech shared task. Their study explores various machine learning techniques to classify sentiments, contributing to the development of sentiment analysis for underrepresented languages.

The research by Sajeetha et al (Thavareesan and Mahesan, 2019) explores various machine learning approaches for sentiment analysis in Tamil texts, including lexicon-based and supervised learning methods. By comparing these techniques, the study identifies effective strategies for feature representation and sentiment classification, highlighting the need for customized methodologies to analyze Tamil sentiments.

The work of Sharmista(Sharmista and Ramaswami, 2020) focuses on the extraction of opinions within Tamil language social media reviews, addressing the scarcity of research in this area. The study aims to improve the understanding of consumer sentiments among Tamil speakers, providing foundational insights that can be leveraged for broader sentiment analysis applications.

Sentiment	Training	Testing
Positive	578	75
Negative	407	46
Neutral	638	70
Opinionated	1316	172
Substantiated	412	51
Sarcastic	790	108
None of the above	172	25

Table 1: Split up of training and testing data into 7 sentiment classes

The research of Ponnusamy et al(Ponnusamy et al., 2023) tackles the complexities of detecting sentiments in code-mixed comments between Tamil and Tulu languages on social media platforms. It proposes preprocessing techniques to improve model performance and offers insights into handling linguistic diversity in sentiment analysis tasks. Hetu et al.(Bhavsar and Manglani, 2019) built and proposed a model in sentiment analysis on twitter data . They classify the emotions based on positive and negative reviews. This model gives high accuracy on large dataset.

3 Dataset Description

The training dataset consists of 4353 Tamil tweets. These are categorized into 7 classes: Positive, Negative, Neutral, Opinionated, Substantiated, Sarcastic, and None of the above. The distribution of the data across these sentiment classes is listed in Table 1. From Table 1, we can infer distribution of data into these classes is not equal.The test set consists 544 Tamil tweets.

4 Proposed Work

The objective of this work is to develop an advanced text classification system capable of categorizing Tamil text into predefined sentiment categories, such as opinionated, neutral, substantiated, positive, sarcastic, and negative. The task involves utilizing machine learning and deep learning techniques to effectively capture the nuances in Tamil text and accurately classify the content based on sentiment.

The first step involves preprocessing the data, which includes extracting text and sentiment labels

from the dataset. The sentiment labels are encoded into numerical values to facilitate the training process. Typically, the labels are transformed using techniques such as Label Encoding or One-Hot Encoding. The dataset is then divided into training and validation sets, with a common split of 80% for training and 20% for validation.

The text processing step includes tokenizing the text, where words are converted into sequences of integers, and then padding or truncating the sequences to a fixed length. This process ensures that all input sequences are of the same length, which is essential for feeding the data into machine learning models.

For model development, various machine learning and deep learning algorithms are explored. Traditional approaches like Naive Bayes and Support Vector Machines (SVM) and LSTM (Long-Short-Term Memory) models are used for this text classification task.

Training of the model involves optimizing hyperparameters such as learning rate, batch size, and the number of epochs to achieve the best performance. The model is then evaluated using metrics like accuracy and Macro F1-score to assess its ability to correctly classify the text into the respective sentiment categories.

Finally, once the model is trained and evaluated, it is used to make predictions on unseen data. The predicted sentiment labels are mapped back to their corresponding categories, and the results are saved in a structured format such as CSV or JSON for further analysis.

5 Experimental Results

The implementation of the proposed approach is carried out in multiple stages, including data preprocessing, feature extraction, model training, and prediction. Initially, the text in the dataset is cleaned and standardized through preprocessing. All text is converted to lowercase to maintain uniformity, and special characters, punctuation, and other non-alphanumeric symbols are removed using regular expressions to reduce noise. Sentiment labels are then encoded into numerical values to facilitate model training. Given the challenges associated with Tamil text, such as spelling variations and code-mixing, careful preprocessing is performed to improve classification accuracy.

For feature extraction, two different techniques are employed: Term Frequency-Inverse Document Frequency (TF-IDF) and Bag-of-Words (BoW) using CountVectorizer. The TF-IDF representation is used to capture important textual features by considering word frequency while reducing the impact of common words, thereby enhancing the model's ability to distinguish between sentiment classes. The BoW model, on the other hand, represents text as a sparse matrix of token counts, providing a straightforward yet effective approach for text classification tasks.

To classify the preprocessed text, two machine learning models—Support Vector Machine (SVM) and Multinomial Naïve Bayes (MNB)—are experimented with. The SVM classifier, using a linear kernel, is trained on TF-IDF features, leveraging its ability to handle high-dimensional text data and provide robust decision boundaries. The MNB model, a probabilistic classifier well-suited for text classification, is trained using BoW features and efficiently processes word frequency-based data. The trained models are then used to predict sentiments on the test dataset, where the same preprocessing and feature extraction techniques are applied to ensure consistency.

Additionally, Long-Short-Term Memory (LSTM) networks are utilized for classifying Tamil text data into various sentiment categories, including opinionated, neutral, substantiated, positive, sarcastic, and negative. The dataset is typically stored in a CSV file and read into a pandas DataFrame for preprocessing. The sentiment labels, representing different categories, are numerically encoded using Label Encoding. This transformation ensures compatibility with machine learning models by converting categorical labels into numeric values.

Subsequently, the dataset is split into training and testing sets using the train-test-split function from scikit-learn. A typical split ratio of 80% for training and 20% for testing is used, ensuring that the model is trained on a large portion of the data while its performance is evaluated on unseen data.

Model	Macro F1-Score	Accuracy
SVM	0.29	0.3566
Naïve Bayes	0.31	0.3566
LSTM	0.2282	0.3056

Table 2: Performance of Training Data

Model	Macro F1-Score	Accuracy
SVM	0.2747	0.35
Naïve Bayes	0.2726	0.35
LSTM	0.2585	0.32

Table 3: Performance on Testing Data

The text data is then preprocessed using a Tokenizer, which converts the text into sequences of integers. These sequences are either padded or truncated to a fixed length, ensuring that all input sequences have the same length. This step is necessary for feeding the data into an LSTM model, which requires a consistent input shape. Padding ensures that shorter sequences are extended, while longer sequences are truncated to maintain uniformity.

The LSTM model is then defined using the Sequential API from Keras. The model consists of an Embedding layer, which converts the input sequences into dense vectors, followed by an LSTM layer that captures temporal dependencies in the text. Dense layers with ReLU activation are included for hidden layers, along with a final Dense layer containing a softmax activation function for multi-class classification. The softmax activation is used to output probabilities for each sentiment class, and the class with the highest probability is selected as the predicted sentiment.

Once the model is trained, it is evaluated using the test set. The test data is tokenized and padded in the same manner as the training data, and the model's predictions are generated. The predicted labels, initially in numeric form, are mapped back to their corresponding sentiment categories using a predefined label mapping. The predictions are then saved to a CSV file, along with the original text and predicted labels.

Finally, the predicted sentiment labels are decoded back into their respective categories and stored for evaluation. The performance of the models is assessed using metrics such as accuracy and macro F1-score, which provide insights into overall classification performance and are tabulated in Table 2. Implementation of these models are available in github.[1]

[1] https://github.com/Dharunika-07/Political_sentiment_analysis

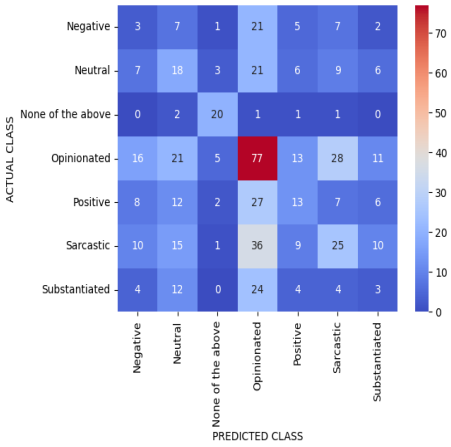

Figure 1: Confusion matrix for performance of SVM on testing dataset

6 Result

The performance of three machine learning models— Support Vector Machine (SVM), Naive Bayes (NB), and Long-Short-Term Memory (LSTM)— was evaluated on the Tamil Political Sentiment Analysis dataset. The results were presented in terms of macro F1-score and accuracy.

A slightly higher macro F1-score (0.2747) was achieved by the SVM model compared to Naive Bayes (0.2726) and LSTM (0.2585), while accuracies of 0.35, 0.35, and 0.32 were attained, respectively. The performance can be viewed in Table 3 and the confusion matrix for SVM is available in Figure 1. From the confusion matrix (Figure 1), we can say that the number correctly predicted sentences is higher for Opinionated class.

The model struggles to identify content that is neutral or factual and tends to assume it is opinion-based. This indicates that the model finds it challenging to distinguish between objective information and subjective language. Additionally, the Substantiated category was frequently confused with Neutral and Opinionated, suggesting that the model may have difficulty recognizing content that is supported by evidence.

7 Limitations

SVM is best suited for binary classification. Handling 7 classes requires one-vs-one or one-vs-all, which increases computation and may cause class imbalances. It also struggles with non-linear text relationships. SVM struggles if emotions in text are non-linearly separable. Emotions like sarcasm require deeper contextual understanding, which SVM lacks.

Naïve Bayes assumes that words occur independently, which is not true for Tamil. Tamil sentences have agglutination (words are formed by joining morphemes), affecting the assumption. It also struggles with classification between sarcasm neutrality.

Training LSTMs for Tamil takes time and GPU power. Tamil has long, context-dependent sentences, which may not be fully captured by a simple LSTM. It also needs large labeled data and requires expensive training.

8 Conclusions

In this task, we have secured the 15^{th} position. In this paper, a machine learning approach for Political Multiclass Sentiment Analysis of Tamil tweets is presented. SVM, Naïve Bayes, and LSTM models were applied, all demonstrating similar accuracy, with SVM slightly outperforming in terms of macro F1-score. Despite challenges such as code-mixing and limited annotated data, Tamil political sentiments were effectively classified using these methods. Future work could involve the exploration of deep learning techniques and data augmentation to address class imbalance and improve accuracy. This study contributes to sentiment analysis in Indian languages and provides insights for analyzing political discourse on social media.

References

Hetu Bhavsar and Richa Manglani. 2019. Sentiment analysis of twitter data using python. *International Research Journal of Engineering and Technology (IRJET)*, 6(3):510–511.

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.

Pranav Gunhal. 2023a. Analyzing political sentiment of indic languages with transformers.

Pranav Gunhal. 2023b. Sentiment analysis in indian elections: unravelling public perception of the karnataka elections with transformers. *International Journal of Artificial Intelligence & Applications*, 14(5):41–55.

V Gokula Krishnan, Pinagadi Venkateswara Rao, J Deepa, and V Divya. Twitter sentiment analysis using ensemble classifiers on tamil and malayalam languages.

Kishore Kumar Ponnusamy, Charmathi Rajkumar, Prasanna Kumar Kumaresan, Elizabeth Sherly, and Ruba Priyadharshini. 2023. Vel@ dravidianlangtech: Sentiment analysis of tamil and tulu. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*, pages 211–216.

M Rajasekar and Angelina Geetha. 2023. Sentiment analysis of tamil tweets using deep convolution neural networks. In *2023 First International Conference on Advances in Electrical, Electronics and Computational Intelligence (ICAEECI)*, pages 1–5. IEEE.

A Sharmista and Dr M Ramaswami. 2020. Sentiment analysis on tamil reviews as products in social media using machine learning techniques: A novel study. *Madurai Kamaraj University Madurai-625*, 21.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2019. Sentiment analysis in tamil texts: A study on machine learning techniques and feature representation. In *2019 14th Conference on industrial and information systems (ICIIS)*, pages 320–325. IEEE.