Team_Catalysts@DravidianLangTech 2025: Leveraging Political Sentiment Analysis using Machine Learning Techniques for Classifying Tamil Tweets

Kogilavani Shanmugavadivel[1], **Malliga Subramanian**[2], **Subhadevi K**[1], **Sowbharanika Janani J S**[1], **Rahul K**[1]
[1]Department of AI, Kongu Engineering College, Perundurai, Erode.
[2]Department of CSE, Kongu Engineering College, Perundurai, Erode.
{kogilavani.sv, mallinishanth72}@gmail.com
{subhadevik,sowbharanikajananijs,rahulk}.22aid@kongu.edu

Abstract

This work proposed a methodology for assessing political sentiments in Tamil tweets using machine learning models. The approach addressed linguistic challenges in Tamil text, including cleaning, normalization, tokenization, and class imbalance, through a robust preprocessing pipeline. Various models, including Random Forest, Logistic Regression, and CatBoost, were applied, with Random Forest achieving a macro F1-score of 0.2933 and securing 8th rank among 153 participants in the Codalab competition. This accomplishment highlights the effectiveness of machine learning models in handling the complexities of multilingual, code-mixed, and unstructured data in Tamil political discourse. The study also emphasized the importance of tailored preprocessing techniques to improve model accuracy and performance. It demonstrated the potential of computational linguistics and machine learning in understanding political discourse in low-resource languages like Tamil, contributing to advancements in regional sentiment analysis.

Keywords: Sentiment Analysis, Machine Learning, Tamil Tweets, Political Sentiments, Random Forest, Class Imbalance, Natural Language Processing (NLP), Tokenization, Computational Linguistics, Multilingual Sentiment Analysis.

1 Introduction

Sentiment analysis, a key area of natural language processing (NLP), is vital for understanding public opinion, especially on social media platforms like Twitter. It helps monitor shifts in sentiment, offering insights into political discourse and its implications for governance.

This study focuses on sentiment analysis of Tamil tweets, presenting unique challenges due to Tamil's complex script, informal idioms, and limited annotated datasets. Traditional techniques, designed for languages like English, often fall short, highlighting the need for language-specific approaches.

Previous studies, such as Mutanov et al. (2021), addressed imbalanced sentiment classes using resampling techniques, with logistic regression, decision trees, and random forests showing strong performance. Liu et al. (2017) highlighted the effectiveness of SVM in sentiment classification, emphasizing feature selection, while Bouazizi and Ohtsuki (2018) introduced "quantification" to capture multiple sentiments within a single post.

This work employs machine learning models like Random Forest, Logistic Regression, Naive Bayes, Decision Tree, AdaBoost, Gradient Boost, and CatBoost to classify Tamil tweets as positive, negative, or neutral. Techniques such as text normalization, stop-word removal, and tokenization with Stanza's Tamil NLP model were applied, contributing to the advancement of multilingual sentiment analysis, particularly for regional languages like Tamil.

2 Related Works

Political sentiment analysis became an important tool for assessing public opinion, particularly in political contexts. Elghazaly et al. (2016) employed SVM and Naïve Bayes to classify Twitter data during Egypt's 2012 presidential election. The study used Term Frequency-Inverse Document Frequency (TF-IDF) for vectorization and found that Naïve Bayes outperformed SVM in terms of accuracy and error rates. Similarly, Bose et al. (2019) utilized sentiment analysis with the NRC Emotion Lexicon and ParallelDots AI APIs to monitor the 2017 Gujarat Legislative Assembly Election. Their methodology categorized tweets as positive, negative, or neutral, effectively summarizing public sentiment. In contrast, Singhal et al. (2015) focused on a context-aware, semantics-based approach to predict election results by analyzing Twitter data from

Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 244–248

the 2019 Indian General Election. They proposed a rules-based system to extract sentiment, which aligned with actual election outcomes, highlighting the importance of domain-specific techniques in political sentiment analysis.

More advanced techniques, such as Long Short Term Memory (LSTM) networks, were used to improve sentiment classification accuracy. For example, Ansari et al. (2020) employed LSTM to analyze Twitter data from the 2019 Indian General Elections. They evaluated the performance of LSTM against classical machine learning models and found that deep learning models, particularly LSTM, outperformed traditional methods. Similarly, Pinto and Murari (2019) investigated the use of LSTM for real-time political sentiment analysis by evaluating tweets about the Ayodhya dispute. Their findings demonstrated that LSTM effectively tackled challenges presented by multiple languages and large datasets. Desai and Mehta (2016) compared various sentiment analysis algorithms, including Naïve Bayes, SVM, and neural networks. Their survey highlighted the benefits of deep learning models and demonstrated their efficiency in categorizing unstructured Twitter data as positive, negative, or neutral.

In addition to these strategies, hybrid approaches combining lexicon-based and machine learning techniques were shown to improve sentiment classification accuracy. For example, Ringsquandl and Petkovic (2013) proposed a hybrid strategy for improving aspect extraction by combining noun phrase frequency and Pointwise Mutual Information (PMI), resulting in higher sentiment classification accuracy. Similarly, Thavareesan and Mahesan (2019) tested numerous sentiment analysis techniques on Tamil literature, including lexicon-based, machine learning, and hybrid methods. Their research revealed that the supervised machine learning methodology, which used fastText and customized corpora, outperformed other methods, achieving 0.79% accuracy. Furthermore, Anish and Sumathy (2022) focused on Tamil political evaluations, employing an SVM approach to address language-specific problems such as noise and sarcasm and proposed improvements through context-based sentiment extraction.

Traditional machine learning methods, such as Naïve Bayes and SVM, offered quick and interpretable solutions, while deep learning methods, such as LSTM, provided higher accuracy. The evolution of political sentiment analysis from traditional to deep learning and hybrid approaches reflected the growing sophistication in the field. Each methodology had its strengths, depending on the dataset's complexity, language, and sentiment classification granularity.

3 Problem and System Description

The system was designed to perform sentiment analysis on Tamil political tweets, which presented challenges due to their brief and unstructured nature, mixed-language usage, and the complexity of Tamil. Tweets often used code-switching between Tamil and English, making sentiment analysis more difficult. The goal was to effectively characterize political sentiments in Tamil tweets, addressing challenges such as data imbalance and the unique features of Tamil political discourse.

Sentiment analysis of Tamil political tweets was critical for determining public opinion on political issues. Social media platforms like Twitter (X) served as important sources of real-time political conversation, but research in this area, particularly for Tamil, was limited. This approach aimed to bridge gaps in multilingual sentiment analysis, especially for Dravidian languages.

The system employed a structured pipeline that began with data gathering from Tamil political tweets, followed by pre-processing to address language-specific issues. Stanza was utilized for tokenization, and TF-IDF was used to extract features and identify relevant words. The pre-processed dataset was then used to train various machine learning models for sentiment classification.

This study leveraged the dataset introduced in Ravikiran et al. (2022) on Dravidian Code-Mixed Offensive Span Identification, which is crucial for addressing language-specific challenges in sentiment analysis. Their dataset has significantly contributed to advancing the field of Dravidian language processing Chakravarthi et al. (2025).

3.1 Dataset Description

The dataset consists of Tamil political tweets with content and labels columns, and the labels reflect one of seven sentiment categories as shown in Table 1 below.

The dataset is divided into training, validation, and testing sets. Table 2 illustrates the distribution of the data.

Table 1: Labels

Categories
Substantiated
Sarcastic
Opinionated
Positive
Negative
Neutral
None of the above

Table 2: Dataset Description

Dataset	No. of Tweets
Train	4352
Validation	544
Test	544

4 Methodology

The following methodology describes the stages required in sentiment analysis of political tweets using several machine learning models. The approach consists of three major components: diagrammatic representation, preprocessing stages, and test data predictions.

4.1 Diagrammatic Representation of Proposed Work

The figure 1 below depicts the full sentiment analysis procedure. The process starts with data collection and progresses through preprocessing, dataset balance, feature extraction, model training, evaluation, and prediction on test data. This end-to-end procedure guarantees that raw text data is handled, models are properly trained, and predictions are produced on previously unknown data.

4.2 Preprocessing Steps

This stage converts raw textual data into a format appropriate for model training. The first step in preprocessing is text cleaning. The text is normalized by deleting unnecessary Unicode characters, maintaining only the required script, and removing special characters and numerals. Commonly used spoken versions are normalized (for example, similar-sounding words are replaced), and specific characters are mapped to their basic forms.

Tokenization is then performed using the proper tokenization library. This transforms each sentence into a list of tokens for further processing. Class balancing is also used, which involves upsampling

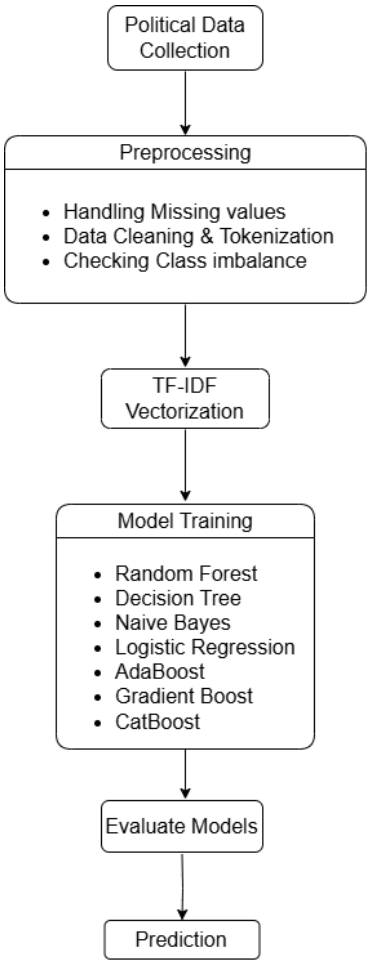

Figure 1: Proposed system pipeline

underrepresented classes to ensure equal representation of all sentiment categories.

4.3 Predictions on Test Data

After the models have been trained, predictions are made using the test dataset. To guarantee consistency, the test data goes through the same preparatory stages (normalization, text cleaning, and tokenization). The TF-IDF vectorizer that was fitted to the training data is utilized to convert the test data into numerical feature vectors.

The trained models are then used to estimate the sentiment of the test data. For example, the Random Forest model is used to assign tweets to one of seven attitude categories: substantive, sarcastic, opinionated, positive, negative, neutral, or none of the above. Once predictions are created, they are saved in a separate column of the test dataset for further examination.

5 Results

The machine learning models performance is measured using measures such as accuracy, precision, recall, and F1-score. During training and testing, each model demonstrated the following accuracies, as shown in Table 3. The code used for preprocessing and analysis can be found in this GitHub repository: Political Multiclass Sentiment Analysis

Table 3: Tarin and Test Accuracy

Model	Train	Test
Random Forest	0.94	0.84
Decision Tree	0.94	0.83
CatBoost	0.70	0.61
Gradient Boost	0.68	0.58
Logistic Regression	0.62	0.52
Naive Bayes	0.57	0.50
AdaBoost	0.28	0.29

To evaluate the performance of various machine learning models in classifying attitudes in political tweets, we analyzed them using key metrics such as accuracy, precision, recall, and F1-score. The results from different models were compared and visualized in the figures below.

Figure 2 presented a comparative analysis of the accuracy of multiple classification algorithms, including Random Forest, Decision Tree, CatBoost, Gradient Boost, and Logistic Regression. The Random Forest model achieved the highest accuracy (0.84), followed closely by the Decision Tree (0.83). CatBoost, Gradient Boost, and Logistic Regression demonstrated lower accuracy levels, indicating their limited effectiveness for this task.

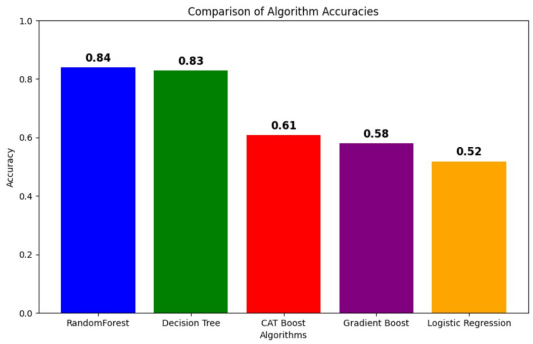

Figure 2: Comparision of Algorithm Accuracies

Figure 3 provided a detailed classification report for the best-performing model, Random Forest, with an overall test accuracy of 83.99%. The model's precision, recall, and F1-score were reported for each sentiment category, demonstrating strong performance in distinguishing between various political attitudes. The confusion matrix further highlighted the model's effectiveness in correctly classifying instances while showing potential misclassifications across specific categories.

```
Test Data Evaluation:
Accuracy: 0.8399790136411333
Classification Report:
                    precision    recall  f1-score   support

          Negative       0.90      0.87      0.89       272
           Neutral       0.85      0.76      0.80       272
None of the above       0.91      0.98      0.94       272
       Opinionated       0.73      0.71      0.72       273
          Positive       0.85      0.87      0.86       272
         Sarcastic       0.84      0.83      0.84       272
     Substantiated       0.80      0.86      0.83       273

          accuracy                           0.84      1906
         macro avg       0.84      0.84      0.84      1906
      weighted avg       0.84      0.84      0.84      1906

Confusion Matrix:
 [[237   3   4   7   2   8  11]
 [  5 208   2  18  14   7  18]
 [  0   0 267   0   0   2   3]
 [ 11  15   6 193  18  21   9]
 [  4   6   6  11 236   1   8]
 [  2   8   5  18   4 226   9]
 [  3   6   4  19   4   3 234]]
```

Figure 3: Classification Report and Confusion Matrix

These results indicated that ensemble models like Random Forest and Decision Tree outperformed other approaches in classifying political sentiments. Their high accuracy and balanced precision-recall scores made them suitable choices for this task. However, further improvements could have been made by optimizing hyperparameters or incorporating advanced deep learning techniques.

6 Conclusion

This study focused on classifying political emotions in Tamil tweets using machine learning models. Ensemble methods like Random Forest and Decision Tree achieved high accuracy by capturing complex patterns, while CatBoost and Gradient Boosting showed promising results. Simpler models like Logistic Regression and Naive Bayes struggled with data complexity, and AdaBoost highlighted the need for more robust models. These findings demonstrate the effectiveness of ensemble techniques in handling linguistic nuances and data imbalances. The study underscores the importance of advanced Machine Learning techniques in political sentiment analysis and suggests exploring deep learning or hybrid models for improved accuracy and deeper insights.

References

Mohd Zeeshan Ansari, Mohd-Bilal Aziz, MO Siddiqui, H Mehra, and KP Singh. 2020. Analysis of political sentiment orientations on twitter. *Procedia computer science*, 167:1821–1828.

Rajesh Bose, Raktim Kumar Dey, Sandip Roy, and Debabrata Sarddar. 2019. Analyzing political sentiment using twitter data. In *Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2018, Volume 2*, pages 427–436. Springer.

Mondher Bouazizi and Tomoaki Ohtsuki. 2018. Multi-class sentiment analysis in twitter: What if classification is not the answer. *IEEE access*, 6:64486–64502.

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.

Mitali Desai and Mayuri A Mehta. 2016. Techniques for sentiment analysis of twitter data: A comprehensive survey. In *2016 international conference on computing, communication and automation (ICCCA)*, pages 149–154. IEEE.

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*, pages 1–5.

Yang Liu, Jian-Wu Bi, and Zhi-Ping Fan. 2017. Multi-class sentiment classification: The experimental comparisons of feature selection and machine learning algorithms. *Expert Systems with Applications*, 80:323–339.

Galimkair Mutanov, Vladislav Karyukin, and Zhanl Mamykova. 2021. Multi-class sentiment analysis of social media data with machine learning algorithms. *Computers, Materials & Continua*, 69(1).

Joylin Priya Pinto and Vijaya Murari. 2019. Real time sentiment analysis of political twitter data using machine learning approach. *International Research Journal of Engineering and Technology (IRJET)*, 6(4):4124–4129.

Manikandan Ravikiran, Bharathi Raja Chakravarthi, Anand Kumar Madasamy, Sangeetha Sivanesan, Ratnavel Rajalakshmi, Sajeetha Thavareesan, Rahul Ponnusamy, and Shankar Mahadevan. 2022. Findings of the shared task on Toxic Span Identification in Tamil. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Martin Ringsquandl and Dusan Petkovic. 2013. Analyzing political sentiment on twitter. In *2013 AAAI Spring Symposium Series*.

Kartik Singhal, Basant Agrawal, and Namita Mittal. 2015. Modeling indian general elections: sentiment analysis of political twitter data. In *Information Systems Design and Intelligent Applications: Proceedings of Second International Conference INDIA 2015, Volume 1*, pages 469–477. Springer.

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.