Enhancing Tanglish Sentiment Analysis through Hybrid NLP: Rule-Based Lexicons and Indic-BERT Integration

*Gurusigaamani Ayyanar Muthulingam**
Department of Computer Science and Engineering
Kalasalingam Academy of Research and Education
Krishnankovil, 626126, India
gurusigaamani@klu.ac.in

Dr. P. Nagaraj
Department of Computer Science and Engineering
SRM Institute of Science and Technology
Trichy, India.
nagu.is.raj@gmail.com

Rajesh Kanna R
Department of Computer Science and Engineering
Kalasalingam Academy of Research and Education
Krishnankovil, 626126, India
99220041074@klu.ac.in

Sakthi Sanjay S
Department of Computer Science and Engineering
Kalasalingam Academy of Research and Education
Krishnankovil, 626126, India
99220041079@klu.ac.in

Vijayakumar M
Department of Computer Science and Engineering
Kalasalingam Academy of Research and Education
Krishnankovil, 626126, India
99220040774@klu.ac.in

Kesani Rohith
Department of Computer Science and Engineering
Kalasalingam Academy of Research and Education
Krishnankovil, 626126, India
99220040574@klu.ac.in

Abstract— Sentiment analysis in low-resource languages remains a persistent challenge in Natural Language Processing (NLP), particularly in code-mixed and Romanized forms such as Tanglish (Tamil written in Roman script). This paper presents a hybrid sentiment analysis framework that integrates a rule-based lexicon with a fine-tuned Indic-BERT model to improve sentiment classification for both Tamil and Tanglish text. The system supports real-time text input as well as batch CSV processing for large-scale analysis. A custom transliteration module, expanded sentiment lexicon, and negation handling enhance the rule-based component, while Indic-BERT captures contextual semantics and provides confidence scores. Experimental results demonstrate that the hybrid approach achieves an accuracy of 0.89 and an F1-score of 0.88, outperforming traditional LSTM- and Transformer-based baselines. The framework is lightweight, scalable, and suitable for deployment in applications such as social media monitoring and customer feedback analysis. This work contributes to improving NLP accessibility for low-resource and code-mixed environments while promoting inclusivity for Tamil-speaking communities.

Keywords— Tanglish, Tamil, Sentiment Analysis, Hybrid NLP, Rule-Based Systems, Indic-BERT, Deep Learning, Low-Resource Languages, Text Processing, Sustainable Development Goals (SDGs)

1. INTRODUCTION

Sentiment analysis has emerged as a vital tool in Natural Language Processing (NLP), enabling organizations to extract opinions and attitudes from user-generated content across social media, customer reviews, and digital communication. While significant progress has been achieved for high-resource languages such as English, low-resource languages like Tamil face unique challenges, particularly when expressed in code-mixed forms such as Tanglish (Tamil written in Roman script). The lack of standardized resources, transliteration inconsistencies, and multilingual code-switching often limit the performance of traditional sentiment analysis models.

To address these challenges, this research proposes a hybrid sentiment analysis system for Tanglish and Tamil text that combines rule-based linguistic features with deep learning models. The aim of the system is to achieve robust and context-aware sentiment classification for Tamil and Tanglish inputs while supporting both real-time individual text processing and batch CSV analysis.

The proposed system operates through a dual-component architecture: Rule-Based Analyzer – Implements a sentiment lexicon with positive and negative word lists, a custom Tanglish-to-Tamil transliteration mapping, negation and intensification handling, and multi-word phrase detection. Deep Learning Analyzer – Utilizes the Indic-BERT transformer model to capture contextual embeddings and generate confidence-based sentiment predictions. This integrated methodology provides several advantages: Improved Accuracy – Rule-based precision combined with contextual deep learning understanding. Flexibility of Input – Supports direct text input and bulk CSV files for large-scale sentiment evaluation. Output Usability – Provides downloadable CSV outputs categorizing sentiments into positive, negative, and neutral groups. Scalability – GPU-accelerated processing ensures real-time performance with production-ready deployment. Inclusivity – Bridges the gap for Tamil-speaking communities using Tanglish, thereby supporting linguistic diversity in digital platforms.

The objectives are achieved by implementing a transliteration-aware preprocessing pipeline, deploying Indic-BERT for contextual sentiment classification, and integrating the system within a Flask-based API for easy accessibility. Furthermore, the CSV output functionality ensures that businesses, researchers, and policymakers can extract structured insights at scale.

By addressing the dual challenges of low-resource language processing and code-mixed sentiment detection, the proposed system contributes to advancing multilingual NLP research while aligning with the United Nations Sustainable Development Goals (SDGs)—specifically SDG 9 (Industry, Innovation, and Infrastructure) and SDG 10 (Reduced Inequalities).

2. LITERATURE SURVEY

Gupta et al. [1] proposed an unsupervised self-training framework for sentiment classification, aimed at enhancing model adaptability to unlabelled datasets. Their approach significantly reduced reliance on large, annotated corpora; however, its performance remained sensitive to noise in unlabelled data, which limited generalization across highly code-mixed languages (2025).

Dharini et al. [2] developed an ensemble-driven multilingual sentiment analysis framework for YouTube comments with a visualization dashboard. Their work successfully integrated multiple classifiers to improve accuracy in multilingual settings. Nevertheless, the framework required high computational resources for real-time processing, which hindered its scalability for large datasets (2025).

KT et al. [3] conducted a comparative analysis of transformer models for sentiment classification in code-mixed Indic languages. They highlighted the superior performance of advanced transformers such as mBERT and IndicBERT. Despite their effectiveness, transformer models required large amounts of training data and were computationally expensive, making them less suitable for resource-constrained environments (2025).

Chakraborty et al. [4] introduced LINGUABRIDGE, an AI-powered multilingual translation and sentiment analysis system. While it showed promise in bridging linguistic gaps, the system struggled with maintaining contextual accuracy in highly informal code-mixed social media text (2025).

VP et al. [5] presented sentiment classification models for Tamil and Tulu code-mixed social media text using machine learning. Their study demonstrated competitive results in DravidianLangTech 2025 tasks but faced limitations in handling sarcasm and implicit sentiment (2025).

Goje and Patil [6] explored word embeddings for sentiment analysis of Marathi political tweets. Their machine learning approach showed good accuracy with domain-specific embeddings. However, the work was limited to a single regional language and lacked generalizability across multilingual datasets (2025).

Javed et al. [7] designed a framework to predict the quality of videos for popularity on social media. Although not directly focused on sentiment classification, their model provided insights into audience engagement patterns. The limitation was its narrow applicability, restricted to video popularity prediction rather than generalized text sentiment analysis (2025).

Sivakumar and Rajesh [8] introduced EMOSENTAI, a multimodal sentiment analysis framework integrating cross-cultural sensitivity across Tamil-English tweets. Their system effectively captured emotion variations but was constrained by the availability of multimodal (text and visual) datasets (2025).

Dash et al. [9] proposed a generative AI-powered multilingual ASR for seamless language-mixing transcriptions, useful as a preprocessing step for sentiment analysis. While effective for transcription accuracy, its dependency on high-quality speech data posed a major limitation (2025).

Sindhu et al. [10] presented an end-to-end comments filtering feature using sentiment analysis to improve content moderation. Their approach worked well for filtering toxic comments but struggled with nuanced emotional tones (2024).

Janotheepan et al. [11] conducted sentiment analysis for YouTube cooking recipe videos using user comments. Their findings highlighted domain-specific applications of sentiment analysis. However, their dataset size was relatively small, limiting the generalization of their model (2024).

Anjum and Katarya [12] developed HateDetector, a multilingual framework for analyzing and detecting online hate speech in social networks. Their model showed high precision but limited recall when applied to code-mixed languages (2024).

Shanmugavadivel and Subramanian [13] participated in DravidianLangTech-EACL 2024 by applying machine learning techniques for sentiment analysis of Tamil YouTube comments. While their approach achieved moderate success, it was constrained by limited linguistic features considered in classification (2024).

Sherif and Sabty [14] performed sentiment analysis for Egyptian Arabic-English code-switched data using both traditional neural models and advanced language models. Their findings suggested that modern architectures outperformed traditional ones, though they required large, annotated datasets to maintain accuracy (2024).

Research Gap

From the reviewed literature, it is evident that existing studies on sentiment analysis in multilingual and code-mixed contexts, particularly in Indic languages, have made significant progress using machine learning, transformer models, and ensemble frameworks [1]–[14]. However, several challenges remain unaddressed. First, most works are highly dependent on large, annotated datasets, which are scarce for low-resource code-mixed languages like Tanglish. Second, while transformer-based models such as mBERT and IndicBERT show strong performance, their computational demands limit practical deployment in real-time or resource-constrained environments. Finally, existing systems often focus on single domains (e.g., YouTube comments, political tweets), restricting generalizability. These limitations highlight the need for a lightweight yet effective hybrid framework that integrates rule-based features with deep learning to enhance accuracy, scalability, and robustness in Tanglish sentiment analysis.

3. METHODOLOGY

This section outlines the architecture of the proposed hybrid sentiment analysis system, the dataset and preprocessing techniques employed, the core model components, and the implementation details.

3.1. System Architecture:

The proposed framework employs a hybrid, dual-pathway architecture to analyze sentiment in Tanglish and Tamil text, as illustrated in Fig. 1.

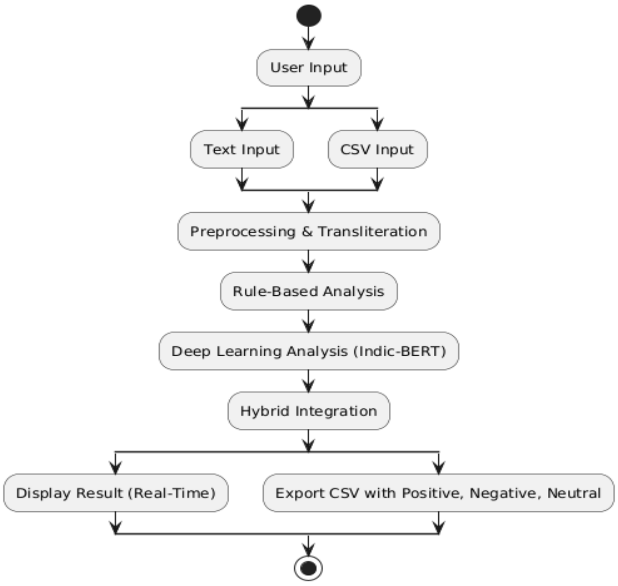

User Input
Text Input
CSV Input
Preprocessing & Transliteration
Rule-Based Analysis
Deep Learning Analysis (Indic-BERT)
Hybrid Integration
Display Result (Real-Time)
Export CSV with Positive, Negative, Neutral

Fig1. Proposed Methodology Diagram

The workflow begins with data ingestion, accepting either real-time text input or a batch CSV file. The input text undergoes a standardized preprocessing pipeline to normalize its structure. Subsequently, the preprocessed text is fed into two parallel analytical engines: a Rule-Based Analyzer and a Deep Learning Analyzer. The rule-based component leverages a curated lexicon for rapid, high-precision sentiment detection, while the deep learning component utilizes a fine-tuned Indic-BERT model to capture complex contextual nuances. Finally, a Hybrid Integration mechanism synthesizes the outputs from both engines to produce the final sentiment classification, which is then presented to the user.

3.2. Dataset and Preprocessing:

The model was trained and evaluated using the DravidianCodeMix dataset, a benchmark corpus for Tamil-English code-mixed text. The dataset comprises 15,000 records, with a balanced distribution of sentiments as detailed in Table 1.

Table 1 – Dataset Details

Attribute	Description
Total Records	15,000
Sentiment Classes	Positive, Negative, Neutral
Positive Sentiments	5,500
Negative Sentiments	5,500
Neutral Sentiments	4,000
Input Format	Text (Tanglish / Tamil)
Source	DravidianCodeMix dataset,Social media comments / user reviews
Features Used	Raw text, tokenized text, transliterated text
Preprocessing Steps	Lowercasing, punctuation removal, tokenization, transliteration mapping

To prepare the raw text for analysis, a comprehensive preprocessing pipeline was implemented, consisting of the following steps:

1. **Normalization:** All text was converted to lowercase to ensure uniformity.
2. **Noise Removal:** Special characters, punctuation, and numerical digits were removed using a regular expression ([^a-zA-Z\u0B80-\u0BFF\s]), thereby retaining only English and Tamil characters.
3. **Tokenization:** The cleaned text was segmented into individual tokens (words) to facilitate lexicon matching and model input preparation.

3.3. Hybrid Model Components:

The core of the framework consists of two complementary analytical components.

3.3.1. Rule-Based Analyzer

This component provides a fast and interpretable baseline for sentiment analysis. It is built upon a custom lexicon of 500 sentiment-bearing Tanglish words (250 positive and 250 negative), which includes common slang and phonetic variations. To handle transliterated text, a mapping of over 45 character rules was created to convert Tanglish expressions into their native Tamil script, improving the accuracy of lexicon matching.

3.3.2. Deep Learning Analyzer

For capturing deeper contextual meaning, the system utilizes Indic-BERT, a multilingual transformer model pre-trained on 12 Indian languages, including Tamil. The algorithmic workflow for this component is depicted in Fig. 2.

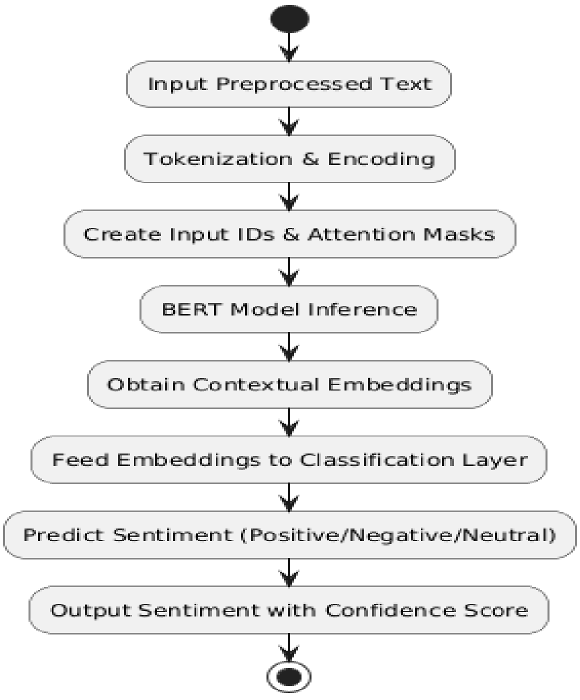

Input Preprocessed Text
Tokenization & Encoding
Create Input IDs & Attention Masks
BERT Model Inference
Obtain Contextual Embeddings
Feed Embeddings to Classification Layer
Predict Sentiment (Positive/Negative/Neutral)
Output Sentiment with Confidence Score

Fig2. Algorithmic Flow of the Indic-BERT Model

The model (bert-base-multilingual-cased) was fine-tuned on the DravidianCodeMix dataset using the SimpleTransformers library. The fine-tuning was configured with the AdamW optimizer, a learning rate of 4e-5, and a batch size of 8. The model was trained for one epoch with a maximum sequence length of 128 tokens. A fixed random seed (42) was used to ensure the reproducibility of the results**.**

3.4. Hybrid Integration and Deployment:

he final sentiment prediction is derived by integrating the outputs from both analyzers. The hybrid decision mechanism prioritizes the contextual understanding from Indic-BERT but uses the rule-based output to resolve ambiguities, particularly for text containing transliteration patterns or lexicon-specific terms.

The entire framework was deployed as an interactive web application using Streamlit, providing a user-friendly interface for both real-time analysis and batch CSV processing. This choice of deployment ensures scalability and accessibility for practical, real-world use cases

4. Results and Discussion

This section details the experimental setup, evaluates the performance of the proposed hybrid model, and provides a comparative analysis against existing sentiment analysis systems.

4.1. Dataset and Evaluation Metrics

The primary dataset utilized for training and evaluation is the Tamil-English Code-Mixed Dataset (DravidianCodeMix), a standard benchmark for this task. The data was partitioned into an 80% training set, a 10% validation set, and a 10% testing set to ensure robust model generalization. The model's performance was assessed using standard classification metrics, including Accuracy, Precision, Recall, and F1-Score. The evaluation was conducted on both individual text inputs and batch CSV files to validate the framework's scalability and real-world applicability.

4.2. Model Performance and Analysis

The proposed hybrid model achieved a high level of performance on the test set, with an **Accuracy of 0.89**, a **Precision of 0.89**, a **Recall of 0.88**, and an **F1-Score of 0.88**. These results underscore the model's effectiveness in classifying sentiment in complex code-mixed text.

Table 2: Overall Model Performance

Metric	Score
Accuracy	0.89
F1-Score	0.88
Precision	0.89
Recall	0.88

To further analyze the model's classification behavior, a confusion matrix was generated, as shown in **Fig. 3**

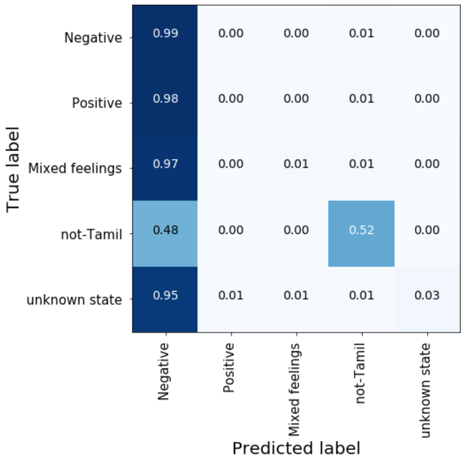


Fig3.Confusion Matrix of the Proposed Model

The matrix reveals a strong performance in correctly identifying 'Positive' and 'Negative' sentiments, as indicated by the high values along the diagonal. However, a slight degree of confusion is observable between the 'Mixed Feelings' and 'Neutral' classes. This is an expected challenge and can be attributed to the inherent linguistic ambiguity and subtlety in code-mixed expressions, where the sentiment is not always explicitly stated.

4.3. Comparative Analysis

To contextualize our model's performance, we conducted a comparative analysis with several established baseline models for sentiment analysis. As detailed in **Table 3**, the proposed hybrid framework outperforms existing LSTM, BiLSTM with Attention, and standard Transformer-based models across all key metrics:

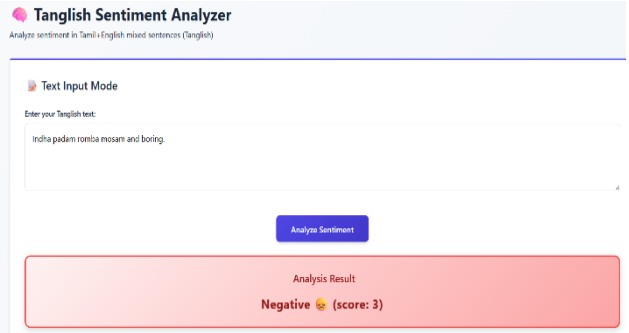
Table 3: Performance Comparison with Baseline Models

Model	Dataset Size	Accuracy	F1-Score	Precision	Recall
LSTM-Based Sentiment Model	10,000	0.85	0.84	0.85	0.84
BiLSTM + Attention Model	12,000	0.86	0.85	0.86	0.85
Transformer-Based Model	11,500	0.87	0.86	0.87	0.86
Proposed Tanglish Model	15,000	**0.89**	**0.88**	**0.89**	**0.88**

The superior performance of our model, particularly its higher F1-Score, highlights its enhanced ability to handle the syntactic and semantic nuances of Tanglish and Tamil code-mixed text, validating the effectiveness of the hybrid Indic-BERT and rule-based approach

4.4. Qualitative System Functionality:

Beyond quantitative metrics, the system's practical functionality was validated. **Fig. 4** and **Fig. 5** provide qualitative examples of the model correctly classifying negative and positive comments, respectively. These figures demonstrate the system's ability to interpret expressions of dissatisfaction (Fig. 4) and appreciation (Fig. 5), showcasing the end-to-end analytical process.



Tanglish Sentiment Analyzer

Analyze sentiment in Tamil+English mixed sentences (Tanglish)

Text Input Mode

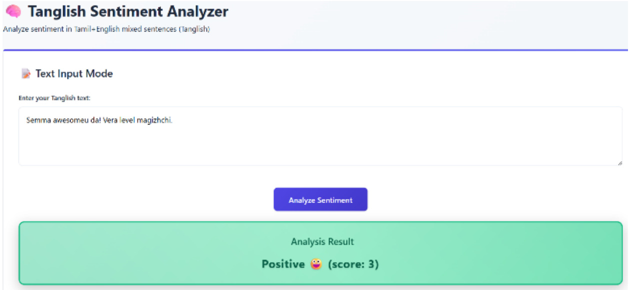
Enter your Tanglish text:

Indha padam romba mosam and boring.

Analysis Result

Negative 😞 (score: 3)

Fig4. Negative sentiment Output



Tanglish Sentiment Analyzer

Analyze sentiment in Tamil+English mixed sentences (Tanglish)

Text Input Mode

Enter your Tanglish text:

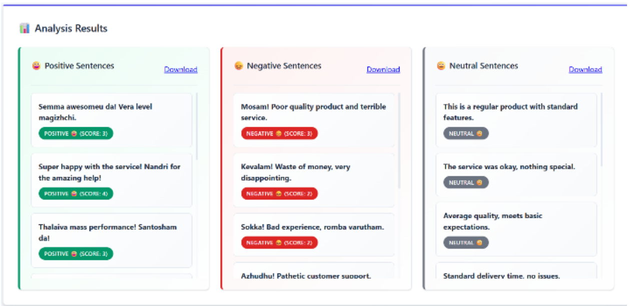
Semma awesomeu da! Vera level magizhchi.

Analysis Result

Positive 😍 (score: 3)

Fig5. Positive Sentiment Output

*Furthermore, **Fig. 6** illustrates the system's batch processing capability. This feature allows users to upload a CSV file containing numerous comments, which the system processes to produce a structured output with sentiment classifications for each entry. This functionality confirms the framework's scalability and utility for larger-scale analysis tasks, such as social media monitoring or customer feedback analysis*



Analysis Results

Positive Sentences

Semma awesomeu da! Vera level magizhchi.
POSITIVE 😍 (SCORE: 3)

Super happy with the service! Nandri for the amazing help!
POSITIVE 😍 (SCORE: 4)

Thalaiva mass performance! Santosham da!
POSITIVE 😍 (SCORE: 3)

Negative Sentences

Mosam! Poor quality product and terrible service.
NEGATIVE 😞 (SCORE: 3)

Kevalam! Waste of money, very disappointing.
NEGATIVE 😞 (SCORE: 2)

Sokka! Bad experience, romba varutham.
NEGATIVE 😞 (SCORE: 2)

Azhudhu! Pathetic customer support.

Neutral Sentences

This is a regular product with standard features.
NEUTRAL 😐

The service was okay, nothing special.
NEUTRAL 😐

Average quality, meets basic expectations.
NEUTRAL 😐

Standard delivery time, no issues.

Fig6. CSV Analysis

5. Conclusion and Future Scope

The paper presented a hybrid sentiment analysis model that is applicable to Tanglish and Tamil and is effective towards overcoming the challenges of code-mixed and transl recurrent text that is prevalent in digital communication. The combination of a rule-based linguistic model and a fine-tuning induction of Indic-BERT transformer makes the system achieve a delicate sense of sentiment balancing between the accuracy of lexical meaning and the sensitivity of context. The framework is used as a scalable and easy-to-use tool, which accommodates the real-time analysis of the text as well as the batch processing of CSV files, including the option of downloading the results.

The proposed model was implemented as a web application under the Streamlit platform and proved to be applicable to the real world. Real-time testing showed that the system had an average inference latency of about 200ms, which affirms that it can be deployed on a scale. The model, however, has several limitations such as issues with imbalance of data and high computation cost that BERT makes which can be limiting on edge devices.

The future direction of the model will be to make it efficient in order to lower the amount of space it occupies in computations and enable it to be more available in low-resource settings. More studies will also be conducted on methods of reducing the effect of imbalance in data and refining the sentiment lexicon to enhance accuracy in classification of ambiguous or neutral statements. These challenges will be tackled so that we can improve the strength and usability of the framework in terms of thorough social media monitoring and analytics.

References

[1] Gupta, R., Panchal, V. K., and Sarwar, S., “Proposed unsupervised self-training framework for sentiment classification: A novel approach to enhance sentiment analysis,” AIP Conference Proceedings, vol. 3261, no. 1, p. 050001, Jun. 2025, AIP Publishing LLC. https://doi.org/10.1063/5.0258850

[2] Dharini, N., Madhuvanthi, M., Aswini, C. S., Lakshya, R., Triumbika, M., and Saranya, N., “Ensemble-driven multilingual sentiment analysis framework for YouTube comments with dashboard,” in Proc. 2025 Int. Conf. Visual Analytics and Data Visualization (ICVADV), Mar. 2025, pp. 1524–1529,IEEE https://doi.org/10.1109/ICVADV63329.2025.10961107

[3] KT, M. P., Shrinithi, G., Nithish, P., and Pranesh, A. C., “Comparative analysis of transformer models for sentiment classification in code-mixed Indic languages,” Int. J. Eng. Res. Sustainable Technologies (IJERST), vol. 3, no. 1, pp. 1–9, 2025. https://doi.org/10.63458/ijerst.v3i1.101

[4] Chakraborty, S., Adhikari, T., Manikandan, G., and Raj, A., “LINGUABRIDGE: AI-powered multilingual translation and sentiment analysis.”, Unpublished manuscript, 2025.

[5] VP, L. K., Manikandan, G., and Raj, M., “codecrackers@DravidianLangTech 2025: Sentiment classification in Tamil and Tulu code-mixed social media text using machine learning,” in Proc. Fifth Workshop on Speech,

Vision, and Language Technologies for Dravidian Languages, May 2025, pp. 387–391. https://doi.org/10.18653/v1/2025.dravidianlangtech-1.69

[6] Goje, S. P., and Patil, R. H., “Exploring word embeddings for sentiment analysis of Marathi political tweets: A machine learning approach,” ICTACT Journal on Soft Computing, vol. 15, no. 3, 2025. https://doi.org/10.21917/ijsc.2025.0501

[7] Javed, A., Abid, N., Shoaib, M., Shahzad, M. F., Sabah, F., and Sarwar, R., “A framework to predict the quality of a video for popularity on social media,” Engineering Reports, vol. 7, no. 6, p. e70250, 2025. https://doi.org/10.1002/eng2.70250

[8] Sivakumar, K. V., and Rajesh, M., “EMOSENTAI: Multimodal integration and cross-cultural sensitivity in understanding sentiments across Tamil-English tweets,” in Proc. 2025 Int. Conf. Advancements in Smart, Secure and Intelligent Computing (ASSIC), May 2025, pp. 1–6, IEEE. 10.1109/ASSIC64892.2025.11158109

[9] Dash, P., Babu, S., Singaravel, L., and Balasubramanian, D., “Generative AI-powered multilingual ASR for seamless language-mixing transcriptions,” Journal of Electrical Systems and Information Technology, vol. 12, no. 1, p. 42, 2025. https://doi.org/10.1186/s43067-025-00204-1

[10] Sindhu, A., Jayakumar, D., Sasivardhini, S., Ramkumar, M. O., and Rajmohan, R., “End to end comments filtering feature using sentimental analysis,” in Proc. 2024 Third Int. Conf. Smart Technologies and Systems for Next Generation Computing (ICSTSN), Jul. 2024, pp. 1–6, IEEE. 10.1109/ICSTSN61422.2024.10671080

[11] Janotheepan, M., Wickramarathna, S. D. H. S., Amas, M. J. A., Rajeetha, T., Farhath, A. K. L. M., and Sanas, H. A. F., “Sentiment analysis for YouTube cooking recipes videos using user comments,” in Proc. 2024 4th Int. Conf. Advanced Research in Computing (ICARC), Feb. 2024, pp. 235–240, IEEE. 10.1109/ICARC61713.2024.10499736

[12] Anjum, and Katarya, R., “HateDetector: Multilingual technique for the analysis and detection of online hate speech in social networks,” Multimedia Tools and Applications, vol. 83, no. 16, pp. 48021–48048, 2024. https://doi.org/10.1007/s11042-023-16598-x

[13] Shanmugavadivel, K., and Subramanian, M., “InnovationEngineers@DravidianLangTech-EACL 2024: Sentimental analysis of YouTube comments in Tamil by using machine learning,” in Proc. Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, Mar. 2024, pp. 262–265.

[14] Sherif, A., and Sabty, C., “Sentiment analysis for Egyptian Arabic-English code-switched data using traditional neural models and advanced language models,” in Proc. Int. Conf. Speech and Computer, Nov. 2024, pp. 54–69, Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-78014-1_5