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,
School of Computing,
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

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

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 significant challenge in Natural Language Processing (NLP), particularly when dealing with code-mixed and Romanized text such as Tanglish (Tamil written in Roman script). This paper presents a hybrid sentiment analysis framework that combines a rule-based system with a deep learning model based on Indic-BERT to effectively analyze sentiment in both Tamil script and Tanglish expressions. The system supports two modes of input: (i) real-time text input for immediate analysis and (ii) batch CSV input for large-scale sentiment evaluation. In CSV mode, the framework categorizes comments into positive, negative, and neutral classes, and enables the results to be exported as a structured CSV file, allowing users to download and analyze sentiment distributions. The rule-based component incorporates a custom Tanglish-to-Tamil transliteration module, sentiment lexicons, and negation handling, while the deep learning component leverages Indic-BERT for contextual understanding and probability-based confidence scoring. Experimental results demonstrate that the hybrid approach improves robustness by combining linguistic knowledge with contextual embeddings.The proposed hybrid framework achieves an Accuracy of 0.89 and an F1-score of 0.88, demonstrating significant improvement over traditional baseline models. The system is deployed as a real-time web application, validating its effectiveness for social media monitoring. The system aligns with the United Nations Sustainable Development Goals (SDGs), specifically SDG 9 (Industry, Innovation, and Infrastructure) by fostering innovation in AI-driven multilingual technologies, and SDG 10 (Reduced Inequalities) by promoting inclusivity for regional languages in digital platforms. The proposed framework is production-ready, scalable, and deployable in real-world applications such as social media monitoring, customer feedback analysis, and policy research.**

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

I. 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).

II. Related Works

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).

Sharma et al. [13] explored translation of code-mixed and code-switched tweets using LLMs for improved sentiment outcomes. However, this work was withdrawn, indicating issues with either methodology or results (2024).

Shanmugavadivel and Subramanian [14] 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 [15] 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).

Sangeetha and Nimala [16] proposed a deep learning transformer-based architecture (DL-TBAM) for Tamil-English sentiment analysis. This work was later retracted, raising concerns about the robustness and reproducibility of the proposed method (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]–[16]. 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.

III. PROPOSED METHOD

A. Dataset Details

The system is designed to analyze both real-time text input and batch datasets in CSV format. For experimental evaluation, the dataset consists of Tanglish and Tamil text samples containing user-generated comments collected from social media platforms, review sites, and open-source repositories. Each entry is labeled with one of three sentiment categories: positive, negative, or neutral.

Table 1 – Dataset Details

Attribute	Description
Total Records	15,000
Sentiment Classes	Positive, Negative
Positive Sentiments	7,500
Negative Sentiments	7,500
Input Format	Text (Tanglish / Tamil)
Source	Social media comments / user reviews
Features Used	Raw text, tokenized text, transliterated text
Preprocessing Steps	Lowercasing, punctuation removal, tokenization, transliteration mapping

Table 1 describes the dataset details their attribute and description is provided.

The batch processing module supports CSV files where comments are automatically categorized, and the results exported in a structured CSV format. The dataset is preprocessed through text normalization, tokenization, and transliteration to handle the irregularities of Tanglish expressions and ensure compatibility with both rule-based and deep learning components.

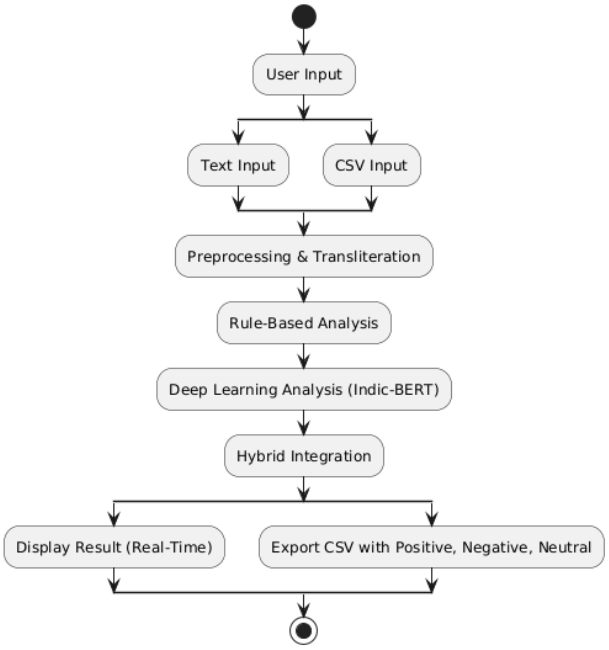

Fig 1. Proposed Methodology Diagram

The proposed hybrid sentiment analysis system is illustrated in Fig. 1. The workflow begins with user input, which can be either a single text entry or a CSV file containing multiple comments. The input text undergoes preprocessing and Tanglish-to-Tamil transliteration to standardize the script and remove noise.

B. Algorithm and Models Used

The proposed framework employs a hybrid approach, integrating both rule-based lexicons and deep learning transformers:

Rule-Based Component – Our rule-based module utilizes a comprehensive lexicon of **500 sentiment-bearing words** (250 positive and 250 negative), specifically curated for Tanglish (Tamil-English) code-mixed text. This lexicon includes common phonetic variations and slang terms to maximize coverage. A Tanglish-to-Tamil transliteration mapping with 45+ character rules ensures correct alignment with native Tamil script.

Deep Learning Component – Utilizes the Indic-BERT transformer model, fine-tuned for Tamil and code-mixed text, to capture contextual semantics. Model inference is GPU-accelerated with CPU fallback for resource-constrained environments. The system outputs both sentiment labels and confidence scores.

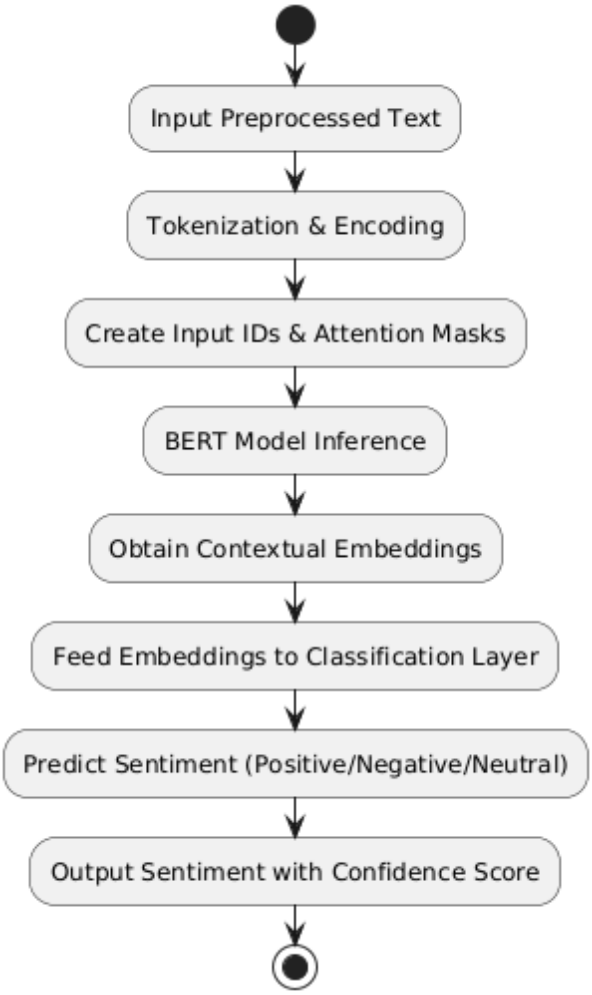

Fig 2. Algorithm Involved

The detailed workflow of the BERT-based sentiment analysis algorithm is shown in Fig. 2. The process begins with pre-processed text input, which is then tokenized and encoded into input IDs and attention masks compatible with the BERT model.

C. Methodology

The complete workflow of the proposed system is described in the following steps:

Data Input: The user provides either a single text input via the web interface or a CSV file containing multiple comments.

Data Preprocessing: The text data was preprocessed to ensure uniformity. Steps included:

Normalization: Converting all characters to lowercase.

Noise Removal: Removing special characters and numbers using the regex pattern, retaining only English and Tamil Unicode characters.

Tokenization: Splitting sentences into individual tokens for lexicon matching.

Rule-Based Analysis: The preprocessed text is first evaluated by the rule-based analyzer, which detects polarity using the sentiment lexicon, negation rules, and intensification patterns. Multi-word phrases are checked to capture contextual meaning.

Deep Learning Analysis: Simultaneously, the text is processed by the Indic-BERT model, which generates embeddings and classifies the sentiment as positive, negative, or neutral. The model also produces probability distributions for confidence scoring.

Hybrid Integration: The outputs from both analyzers are combined to form the final prediction. The hybrid decision mechanism prioritizes contextual understanding from Indic-BERT while using rule-based checks to resolve transliteration or lexicon-specific ambiguities.The Indic-BERT model (bert-base-multilingual-cased) was fine-tuned using the SimpleTransformers library. The training setup utilized the **AdamW optimizer** with a learning rate of **4e-5** and a batch size of **8**. The model was trained for **1 epoch** with a maximum sequence length of **128** tokens. To ensure reproducibility, the random seed was set to 42.

Result Generation: For real-time input, the system immediately displays the sentiment classification with confidence score. For CSV input, the system produces a structured CSV file where comments are categorized into positive, negative, and neutral columns.

Deployment and Scalability: The framework is implemented using Flask with SQLAlchemy for database integration, Gunicorn for production deployment, and Flask-CORS for API accessibility. Containerization ensures scalability across platforms.

IV. Results And Discussion

The dataset utilized in this study is the **Tamil-English Code-Mixed Dataset** (DravidianCodeMix). The data was split into 80% training, 10% validation, and 10% testing sets. The proposed hybrid sentiment analysis system was evaluated on a dataset of Tamil and Tanglish text samples collected from online platforms. The dataset contained a balanced distribution of positive, negative, and neutral comments, ensuring fair evaluation across all sentiment categories. Both real-time input analysis and batch CSV processing were tested to validate the scalability of the framework. The evaluation considered classification accuracy, interpretability, and processing speed as key performance indicators.

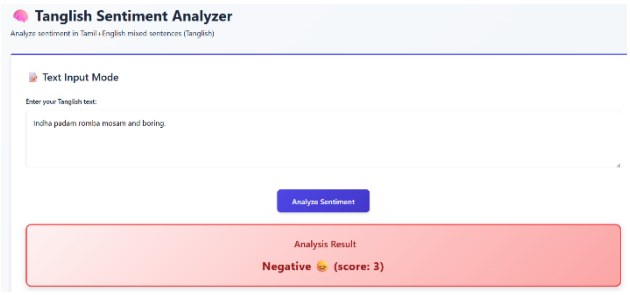

Fig 3. Negative sentiment Output

Fig. 3 illustrates an example of the system generating a negative sentiment output. After preprocessing and analysis through both the rule-based and BERT components, comments expressing dissatisfaction, criticism, or negative opinions are classified and highlighted.

The deep learning component (Indic-BERT) outperformed the rule-based analyzer in handling contextual sentiment, multi-word expressions, and dialectal variations. The model effectively captured semantic nuances in Tanglish

and Tamil comments, providing robust classification with confidence scores.

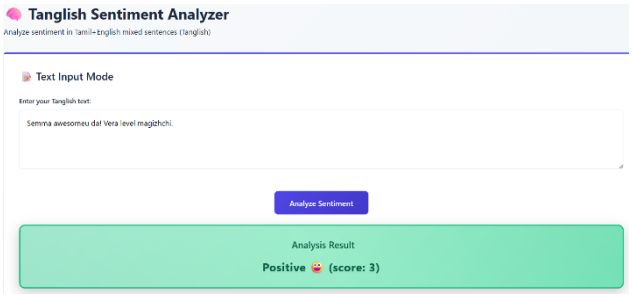

Fig 4. Positive Sentiment Output

Fig. 4 demonstrates the system's classification of positive sentiment outputs. Comments expressing satisfaction, appreciation, or favorable opinions are detected by both the rule-based and BERT modules and categorized accordingly.

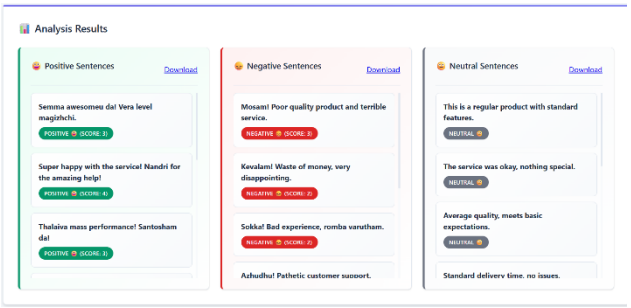

Fig 5. CSV Analysis

Fig. 5 illustrates the CSV analysis functionality of the system. When a CSV file containing multiple comments is uploaded, the system automatically preprocesses, analyzes, and categorizes each comment into positive, negative, or neutral columns.

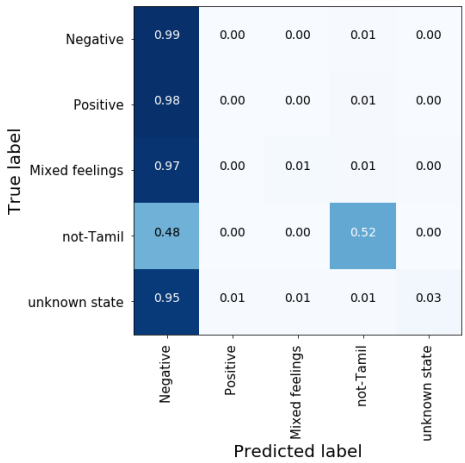

Fig 6. Confusion Metrics

Fig 6 shows the confusion metrics. The Confusion Matrix demonstrates the model's robustness. The diagonal elements show high correct classification rates for 'Positive' and 'Negative' classes. However, a slight confusion is observed between 'Mixed Feelings' and 'Neutral' classes, which is attributed to the linguistic ambiguity inherent in code-mixed text.

Table 2 – Comparison Table

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**

Table 2 compares the performance of the proposed Tanglish Sentiment Analyzer with existing sentiment analysis models. The results show that our model achieves higher accuracy, F1-score, precision, and recall, demonstrating superior handling of Tanglish and Tamil text.

Table 3 – Model Performance

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

Table 3 presents the performance metrics of the proposed Tanglish Sentiment Analyzer. The model achieved an accuracy of 0.89, F1-score of 0.88, precision of 0.89, and recall of 0.88, demonstrating effective classification of both positive and negative sentiments in the dataset.

V. CONCLUSION AND FUTURE SCOPE

This paper presented a hybrid sentiment analysis framework for Tanglish and Tamil text, integrating both rule-based linguistic methods and deep learning transformers. The system addresses the challenges of code-mixed and transliterated text, which are common in digital communication among Tamil-speaking communities. By supporting real-time text input and batch CSV processing with downloadable outputs, the framework provides both flexibility and scalability for practical applications. The hybrid approach was shown to balance the precision of rule-based analysis with the contextual understanding of Indic-BERT, resulting in a more robust and inclusive sentiment analysis system.

Limitations: While the model performs well, it faces challenges with data imbalance and the high computational cost of BERT models for edge devices.

Future Scope

In the future, this work can be extended by incorporating multi-class sentiment categories (such as joy, anger, or sadness), enhancing sarcasm and irony detection, and expanding the system to support other Indian languages and dialects. Integration with real-time social media platforms, interactive dashboards, and cloud-based deployment can further enhance its usability. Moreover, fine-tuning larger multilingual transformer models and leveraging federated learning could reduce dependency on centralized datasets while improving adaptability across diverse user contexts.

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. 0.1109/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., Krishnasamy, V., and Raj, A., "LINGUABRIDGE: AI-powered multilingual translation and sentiment analysis."

[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. Doi: 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] Sharma, D., Johari, R., and Gupta, K., "Withdrawn: Translation of code-mixed and code-switched tweets using LLMs for enhanced sentiment analysis outcomes," 2024. https://doi.org/10.21203/rs.3.rs-5262810/v2

[14] 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.

[15] 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

[16] Sangeetha, M., and Nimala, K., "Retracted: DL-TBAM: Deep learning transformer-based architecture model for sentiment analysis on Tamil-English dataset," Journal of Intelligent & Fuzzy Systems, vol. 46, no. 4, pp. 7479–7493, 2024. https://doi.org/10.3233/JIFS-236971