A Mahendar

Tanglish Research Paper

TURNITIN REPORT

Document Details

Submission ID trn:oid:::3618:124309452	6 Pages
Submission Date Dec 12, 2025, 11:28 AM GMT+5:30	3,588 Words
Download Date Dec 12, 2025, 11:37 AM GMT+5:30	21,438 Characters
File Name Tanglish Research Paper.docx	
File Size 328.0 KB	

0% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

0 AI-generated only 0%
Likely AI-generated text from a large-language model.

0 AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

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
Tiruchirappalli, 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 with low-resource languages is also a significant issue to Natural Language Processing (NLP) in code-mixed and romanized language like Tanglish (Tamil in Roman characters). The given paper also tries to solve this problem by creating a hybrid system of sentiment analysis based on a rule-based sentiment analysis lexicon and a fine-tuned model of Indic-BERT to improve the sentiment analysis classifier of a Tamil and Tanglish text. The service provides predictions on both a real-time text entry basis and batch processing of CSV files. The lexicon of the rule-based sentiment classifier was then enriched and advanced due to the introduction of negation scopes and a custom transliteration module, whereas Indic-BERT was applied to enhance the performance of the rule-based sentiment classifier regarding the contextual semantics and output confidence scores of the predictions. The hybrid sentiment analysis model has a score of 0.89 on accuracy and a score of 0.88 on the F1 score, which is significantly higher than the LSTM and the Transformer-based sentiment analysis model, as the experiments that have been conducted to measure performance indicate. The framework is lightweight, scalable, and applicable to application in social media monitoring and customer feedback analysis. This publication adds to the advancement of NLP accessibility to low-resource and code-mixed settings and inclusivity of Tamil-speaking communities.

Keywords— Tanglish, Tamil, Sentiment Analysis, Hybrid NLP, Rule-Based Systems, Indic-BERT, Deep Learning, Low-Resource Languages.

1. INTRODUCTION

Sentiment analysis has become an essential resource in Natural Language Processing (NLP), allowing organizations to retrieve opinions and attitudes of user-generated content in the field of social media, customer comments, and online communication. Although high-resource languages like English have made major strides, low-resource languages like Tamil have their own special challenges especially when they are written in mixed languages such as Tanglish (written in Roman script). The performance of a traditional sentiment analysis model is often restricted by the absence of standardized resources, inconsistencies in transliteration, and code-switching across languages.

To overcome these problems, the proposed hybrid sentiment analysis framework of both Tanglish and Tamil text in this research involves the combination of linguistic characteristics and rule-based principles on the one hand and deep learning paradigms on the other hand. The objective of the system is to attain strong and context-sensitive sentiment classification of Tamil and Tanglish text inputs and allow real-time individual text processing and overall batch CSV processing.

The suggested system works based on dual-component structure: Rule-Based Analyzer - Refines a sentiment lexicon of positive and negative word lists, a Tanglish-to-Tamil transliteration mapping and negation and intensification processing, and multi-word phrases recognition. Deep Learning Analyzer - This model applies to the Indic-BERT transformer model to identify contextual embeddings and sentiment predictions based on confidence. This hybrid approach offers several benefits: Better Accuracy - rule based accuracy and contextual deep learning insight. Ability to easily add new inputs - Accepts direct text-based inputs and CSV files of large-scale sentiment analysis. Output Usability - Makes CSVs available to be downloaded with the sentiments classified into possible positive, negative, and neutral groups. Scalability - GPU-accelerated processing guarantees that it will be able to run in real time with production-ready deployment being possible. Inclusivity - Closes the divide of Tamil speaking groups who speak Tanglish thus promoting linguistic diversity online.

The above goals are met through transliteration-conscious preprocessing pipeline, using Indic-BERT to classify contextual sentiments, and having the system integrated into a Flask-based API to make it easily accessible. Moreover, CSV output feature also guarantees that businesses, researchers and policy makers can derive structured insights in bulk.

The proposed system can be discussed as the contribution to the existing research in the field of multilingual NLP and its alignment with the United Nations Sustainable Development Goals (SDGs) as SDG 9 (Industry, Innovation, and Infrastructure) and SDG 10 (Reduced Inequalities).

2. LITERATURE SURVEY

Gupta et al. [1] have suggested an unsupervised self-training model to sentiment classification, in order to make the models better adapt to unlabeled data. This strategy contributed to the way they were much less dependent on big, annotated corpora, but the performance of this one was sensitive to noise in unlabeled information, which restricted the generalization of their applications to highly code-mixed languages (2025).

Dharini et al. [2] designed a multilingual sentiment analysis system based on an ensemble on a You Tube comment with a visualization dashboard. Their work managed to combine several classifiers to enhance more accuracy in multilingual environments. However, the framework consumed a lot of computational power to process data in real-time hence limiting its applicability with large datasets (2025).

KT et al. [3] have performed a comparative study of transformer models to classify sentiments in code-mixed Indic languages. They pointed out the higher level of performance of high-tech transformers like mBERT and IndicBERT. Although they worked well, transformer models had high training data requirements and were computationally costly and thus could not be easily deployed in resource-constrained settings (2025).

Chakraborty et al. [4] presented LINGUABRIDGE, which is an AI-based multilingual translator and sentiment analyzer. Though it also presented a promising option of reducing linguistic barriers, the system encountered problems with preserving contextual correctness in highly informal code-mixed social media text (2025).

VP et al. [5] introduced machine learning sentiment classification models of the code-mixed social media text in Tamil and Tulu. Their experiment showed competitive performance on DravidianLangTech 2025 but was limited to sarcasm and implicit sentiment (2025).

Goje and Patil [6] examined word embeddings to carry out sentiment analysis of political Tweets in Marathi. Their machine learning model demonstrated good performance on domain particular embeddings. Nevertheless, it was only done on one regional language and could not be generalized on multilingual datasets (2025).

Javed et al. [7] developed a framework to make predictions on the quality of videos to become popular on social media. Though it is not specifically aimed at the classification of sentiments, their model allowed us to understand the patterns of audience engagement. Its weakness was in its small scope of use, which was only applicable on video popularity prediction, but not on the overall text sentiment analysis (2025).

Sivakumar and Rajesh [8] proposed EMOSENTAI that is a multimodal sentiment analysis framework incorporates cross-cultural sensitivity in Tamil-English tweets. They were successful in capturing emotion variations but limited by access to multimodal (text and visual) datasets (2025).

Dash et al. [9] presented a generative AI-based multilingual ASR to obtain language-mixing transcriptions with ease, which can be used as a pre-processing stage in sentiment analysis. Although it was useful in terms of accuracy in transcription, it was limited significantly by the need to have high quality speech data (2025).

Sindhu et al. [10] provided an analysis to enhance content moderation. It was effective in filtering toxic comments but not so effective with subtle emotional tones (2024).

Janotheepan et al. [11] Their results emphasized domain specific applications of the sentiment analysis. Nevertheless, the size of their dataset was not that large, which restricted the extrapolation of their model (2024).

Anjum and Katarya [12] came up with the HateDetector, a multilingual system based on hate speech analysis and detection in social networks. Their model demonstrated good accuracy and poor recall in the case of code-mixed languages (2024).

Shanmugavadivel and Subramanian [13] also took part in DravidianLangTech-EACL 2024, where they used machine learning to sentimentally analyse Tamil YouTube comments. Their method was moderately successful but limited to use of few language characteristics taken into consideration in classification (2024).

Sherif and Sabty [14] carried out sentiment analysis of Arabic-English code-switched data in Egyptian Arabic language with both traditional neural model and advanced language model. Their results indicated that modern architecture was superior compared to traditional ones, but they needed large datasets that were annotated to remain accurate (2024).

Research Gap

Based on the literature review, it can be seen that the current literature in the field of sentiment analysis in the multilingual and code-mixed setting, especially the Indic languages, has advanced significantly in employing the machine learning frameworks, transformer models, and ensemble models [1]–[14]. But there are several challenges that are not resolved. To begin with, most of the works are strongly reliant on large, annotated datasets, which are not readily accessible to low-resource code-mixed languages such as Tanglish. Second, transformer-based models like mBERT and IndicBERT demonstrate high performance, but their computational requirements restrict their implementation in real-time or resource-constrained systems. Lastly, the available systems usually target one area (e.g. YouTube comments, political tweets) which prevents generalization. The limitations presented show that a lightweight but efficient hybrid system is required, which incorporates the rule-based characteristics with deep learning to improve the accuracy, scaled, and robustness in the Tanglish sentiment analysis.

3. METHODOLOGY

This section describes the architecture of the proposed hybrid sentiment analysis system, used data set and preprocessing methods, the main model blocks and the implementation.

3.1. System Architecture:

The proposed framework has a hybrid, two-way, architecture to examine sentiment in Tanglish and Tamil text as pictured in Fig. 1.

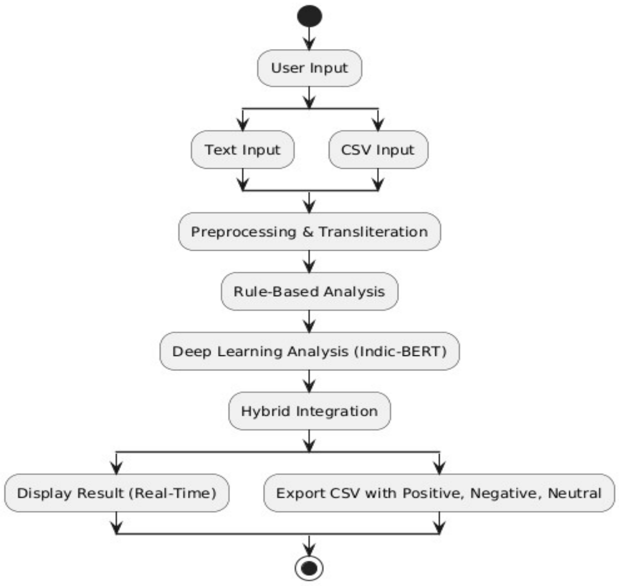

User Input
Text Input
CSV Input
Preprocessing & Transliteration
Rule-Based Analysis
Deep Learning Analysis (Indic-BERT)
Hybrid Integration
Display Result (Real-Time)
Export CSV with Positive, Negative, Neutral

Fig1. Proposed Methodology Diagram

The workflow begins with data entry through real-time or CSV batch file. The inputs are subjected to a common preprocessing pipeline to normalize its structure. The text is then preprocessed and loaded into two parallel engines that analyze the text: a Rule-Based Analyzer and a Deep Learning Analyzer. The rule-based component uses an edited lexicon to detect sentiment quickly and with high accuracy, whereas the deep learning component uses a fine-tuned Indic-BERT model, which can detail the nuances of the context. Lastly, the Hybrid Integration mechanism has the role of synthesizing the results of both engines to obtain the result of the sentiment classification that can be reported back to the user.

3.2. Dataset and Preprocessing:

The model was trained and tested with DravidianCodeMix, a standard corpus of Tamil-English code-mixed text. The records constitute 15,000 samples, and the distribution of the sentiments is balanced as it is indicated 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 to be analysed, a comprehensive preprocessing pipeline was applied and it includes:

1. **Normalization:** To achieve format uniformity, all the text was changed to lower case.
2. **Noise Removal:** A regular expression was used to remove characters with special characters, punctuations, and digits, thereby leaving behind English and Tamil characters.
3. **Tokenization:** The text was removed of any extraneous characters to create tokens representing single words to enable lexicon matching and input model preparation.

3.3. Hybrid Model Components:

The framework comprises two supplementary components of analysis.

3.3.1. Rule-Based Analyzer

The element gives a sentiment analysis a quick and informative baseline. It is founded on a lexicon of custom 500 sentiment-carrying Tanglish words (250 of them positive and 250 of them negative) that comprises frequent slang and phonetic variations. To deal with the transliterated text, a mapping of more than 45 character rules was developed to decode Tanglish expressions into the native Tamil script to make the matching of lexicon more accurate.

3.3.2. Deep Learning Analyzer

To achieve more contextual meaning, the system takes advantage of Indic-BERT, a multilingual transformer model that is trained on 12 Indian languages, including Tamil. The algorithmic representation of the workflow of this component is shown in Fig. 2.

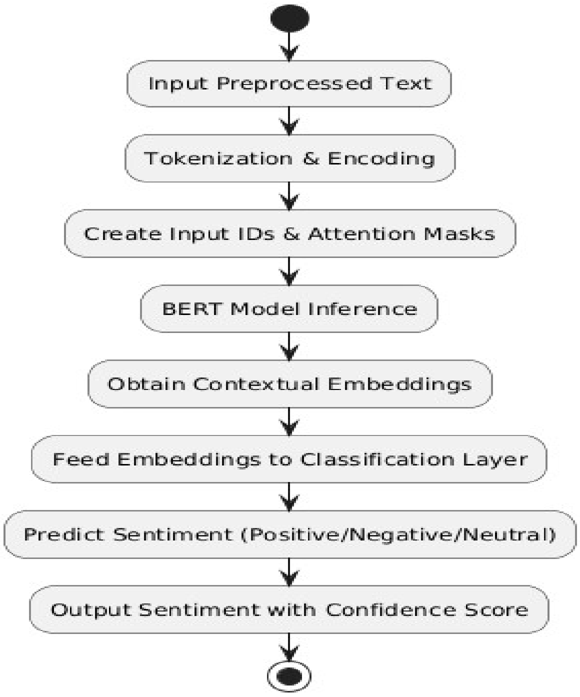

Input Preprocessed Text
Tokenization & Encoding
Create Input IDs & Attention Masks
BERT Model Inference
Obtain Contextual Embeddings
Feed Embeddings to Classification Layer
Predict Sentiment (Positive/Negative/Neutral)
Output Sentiment with Confidence Score

Fig2. Algorithmic Flow of the Indic-BERT Model

The DravidianCodeMix dataset was fine-tuned on the model (bert-base-multilingual-cased) with the Simple Transformers library. The fine-tuning was set up using AdamW optimizer, a learning rate value of 4e-5, and a batch size of 8. The model was fitted to one epoch and the length of maximum sequence is 128 tokens. The results were reproduced to guarantee that the results were reproducible using a fixed random seed (42)**.**

3.4. Hybrid Integration and Deployment:

he the last sentiment prediction is obtained by taking the combination of the outputs of both analysers. The hybrid decision system attaches importance to the contextual knowledge of the Indic-BERT but relies on an output of the rule-based system to address the ambiguities, especially in the case of a text, which contains transliteration patterns or lexicon-specific words.

The whole structure was installed as a web-based interactive application with Streamlit, which is easy to use and can be analyzed real-time and in batches. This deployment option guarantees scalability and accessibility to real-world application.

4. Results and Discussion

This part will describe the experimental design, describe the performance of the proposed hybrid model and give a comparative evaluation of those systems that are available in sentiment analysis.

4.1. Dataset and Evaluation Metrics

The main dataset which is used to both train and do an evaluation is the Tamil-English Code-Mixed Dataset (DravidianCodeMix) which is a common benchmark to this problem. To ensure a strong generalization of the model the data was divided into an 80 percent training set, 10 percent validation set, and 10 percent tests set to achieve a strong generalization of the model. Standard classification metrics were used to evaluate the performance of the model, which are Accuracy, Precision, Recall, and F1-Score. The analysis was performed on an individual input of texts and batch input in the form of CSV files to ensure the scalability and applicability of the framework in real-life scenarios.

4.2. Model Performance and Analysis

The hybrid model proposed performed highly on the test set, having the following Accuracy of 0.89, Precision of 0.89, Recall of 0.88 as well as F1-Score of 0.88. The findings support the usefulness of the model in sentiment classification of complex code-mixed text.

Table 2: Overall Model Performance

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

In order to further examine the classification behavior in the model, a confusion matrix was created as illustrated in Fig. 3

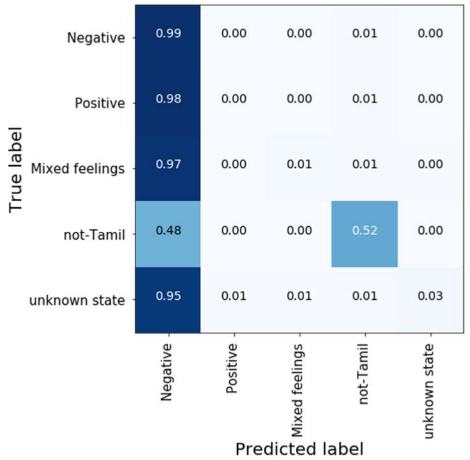


Fig3.Confusion Matrix of the Proposed Model

The matrix depicts high results in the proper identification of the positive and negative sentiment as the results are high along the diagonal. Nevertheless, a little bit of confusion can be noticed between the classes of the Mixed feel feelings and the neutral. This is basically a given challenge, and it can be explained by the linguistic ambiguity and subtlety of code-mixed expressions where the sentiment is not always clearly laid out.

4.3. Comparative Analysis

To put our model into perspective with the existing performance of the models in the sentiment analysis field, we had to do a comparative analysis with some of the renowned baseline models in the field. The proposed hybrid framework is better than the current LSTM, BiLSTM with Attention, and plain Transformer-based models in all the most important metrics, as shown in Table 3:

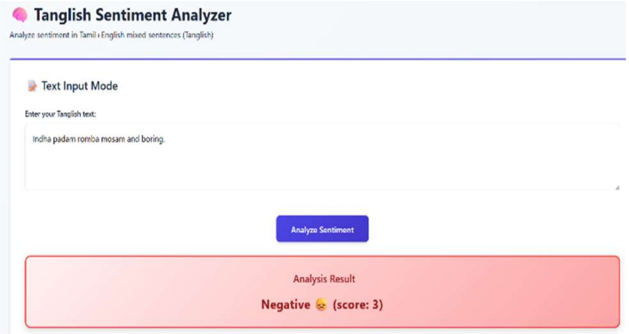
Table 3: Performance Comparison with Baseline Models

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 better score of our model especially higher F1-Score, is evidence that it is better managed to process the syntactic and semantic complexity of the Tanglish and Tamil code-mixed text, confirming the usefulness of the hybrid Indic-BERT and rule-based method.

4.4. Qualitative System Functionality:

In addition to the quantitative measures, practical functionality of the system was tested. Fig. 4 and Fig. 5 give qualitative visualization of the model appropriately categorizing negative and positive comments respectively. These data illustrate the capability of the system to read the signs of dissatisfaction (Fig. 4) and appreciation (Fig. 5), which presents the end-to-end analysis process.

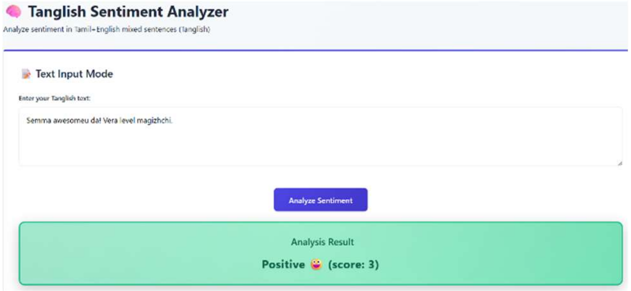

🧠 Tanglish Sentiment Analyzer
Analyze sentiment in Tamil+English mixed sentences (Tanglish)
Text Input Mode
Analyze Sentiment
Analysis Result

Enter your Tanglish text:

Indha padam romba mosam and boring.

Negative 😞 (score: 3)

Fig4. Negative sentiment Output

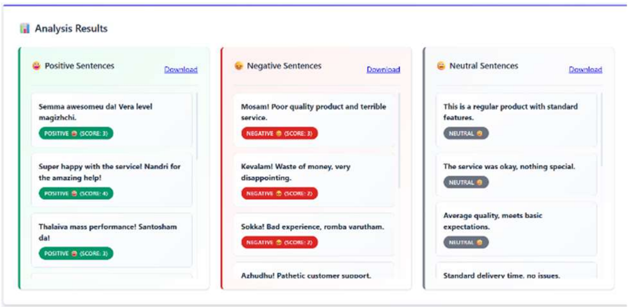

🧠 Tanglish Sentiment Analyzer
Analyze sentiment in Tamil+English mixed sentences (Tanglish)
Text Input Mode
Analyze Sentiment
Analysis Result

Enter your Tanglish text:

Semma awesomeu da! Vera level magizhchi.

Positive 😜 (score: 3)

Fig5. Positive Sentiment Output



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)

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.

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

Moreover, Fig. 6 shows the ability of the system to run in batch. This is what enables one to upload a CSV file with many comments in it, which the system will handle to generate a structured output that will have a sentiment classification on each entry. This functionality attests to the scalability of the framework and its use in the analysis of more large-scale tasks, including social media monitoring or customer feedback 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 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., Krishnasamy, V., 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