International Journal of Science and Technology
ISSN: 2049-7318 | Nov. 2025, Vol. 1, No. 1, pp.1-8.
Doi: https://doie.org/10.97248/IJST.2025558640

SCIRO

Research Paper

Sentiment analysis on Indian movie reviews using a machine learning approach

M.Ramana Reddy
Asst Professor, Department of Computer Science and Applications
Loyola Degree College (YSRR), Pulivendula Y.S.R. Kadapa (District)
Email: ramanareddymca05@gmail.com

D.Himaja
Asst Professor, Department of Computer Science and Applications
Loyola Degree College (YSRR), Pulivendula Y.S.R. Kadapa (District)
Email: devireddyhimaja@gmail.com

KEY WORDS

Sentiment Analysis
Deep Learning
Code Mixing
NLP
Hinglish
BERT
LSTM

Article History

Receives: 26-10.2025
Revised: 10-11.2025
Accepted: 24-11-2025
Published: 29-11-2025

ABSTRACT

The proliferation of digital platforms has intensified the need for accurate sentiment analysis (SA) of user-generated content, particularly concerning Indian cinema. This paper addresses the critical challenge of sentiment classification in Indian movie reviews, which are characterized by significant linguistic heterogeneity, notably *code-mixing* (e.g., Hinglish).[1] The research demonstrates that traditional Machine Learning (ML) approaches, reliant on static features such as Term Frequency–Inverse Document Frequency *(TF-IDF),* are inadequate for resolving the inherent contextual ambiguity, yielding a performance ceiling of approximately 70% F1-score for code-mixed text. To overcome this limitation, a novel *Hybrid Deep Learning Architecture (BERT-Enhanced CNN-LSTM)* is proposed. This model integrates the deep, contextual language understanding of a pre-trained Transformer (BERT), the efficient local feature extraction of Convolutional Neural Networks *(CNNs),* and the long-range dependency modeling of Long Short-Term Memory *(LSTM)* networks. The methodology includes a specialized data preprocessing pipeline focusing on noise reduction and normalization essential for low-resource Indian text.[2] Empirical evaluation on a code-mixed movie review corpus reveals that the proposed Hybrid Model achieves a macro F1-score of *0.9405 (94.05%),* demonstrating significant performance superiority over traditional ML baselines and non-contextual deep learning models. This work contributes a robust and high-accuracy framework essential for effective opinion mining in the diverse South Asian linguistic landscape.

I. INTRODUCTION

A. Context and Commercial Significance of Opinion Mining

The widespread adoption of digital channels has transformed how consumers express their evaluations and grievances concerning products and media. For Indian cinema, online reviews posted on platforms such as blogs, social media, and dedicated databases influence financial success and provide crucial market intelligence.[5] Sentiment analysis *(SA)*, or opinion mining, is the computational field dedicated to extracting and interpreting these expressions of emotion, attitude, and opinion.[6] SA applications extend across various domains, including business, government, and entertainment, underscoring its wide utility. The growing importance of this field coincides directly with the continuous growth of social media, which provides a rich, dynamic source of textual data for analysis.

B. Defining the Challenge: Linguistic Complexity in Indian Reviews

The specific linguistic landscape of India introduces complexities that make sentiment analysis substantially more challenging than in monolingual environments. These challenges fundamentally limit the efficacy of standard *NLP* tools.

The primary difficulty is *Code-Mixing*, which refers to the spontaneous blending of two or more languages, such as Hindi and English, within a single sentence, often utilizing the Roman script for Hindi words (Hinglish).[1] Classifiers designed for structured, monolingual input often fail when encountering this phenomenon because the assumptions underlying grammar and vocabulary consistency are violated. Furthermore, Indian vernacular text is often characterized by linguistic subtleties that require deep contextual awareness. These include complex negation structures ("I wouldn't say the shoes were cheap," implying the opposite sentiment) and the frequent use of culturally specific idioms. Traditional models, which typically rely on keyword counting or fixed semantic representations, struggle to resolve these ambiguities, highlighting the need for context-sensitive machine learning solutions.

A related systemic problem is the scarcity of resources. Many regional Indian languages, including Urdu, Tamil, and Malayalam, are considered low-resource, lacking the large, high-quality, annotated corpora, language parsers, and pre-trained models that are standard for English.[10] This data poverty necessitates the development of specialized preprocessing techniques and architectures capable of extracting maximal information from limited, noisy datasets.

C. Problem Statement and Research Objective

The inability of classical Machine Learning models to adequately process the dynamic and mixed-lingual nature of Indian movie reviews constitutes a significant barrier to accurate opinion mining in this domain. Consequently, the primary research objective is to develop and empirically validate a highly accurate, robust machine learning framework capable of classifying the sentiment polarity *(Positive, Negative, Neutral)* of code-mixed Indian movie reviews. This framework must specifically address the linguistic volatility of Hinglish and demonstrate a clear performance advantage over conventional feature engineering approaches.

D. Paper Contribution

The core contributions of this research are structured to advance the state-of-the-art in code-mixed sentiment analysis:

- **Comparative Benchmarking:** A detailed empirical comparison of the effectiveness of advanced Hybrid Deep Learning architectures against conventional Machine Learning classifiers (*SVM*, Logistic Regression, *etc.*) using metrics relevant to classification imbalance (F1-score, Precision, Recall).
- **Novel Architecture Validation:** The implementation and demonstration of a *BERT-Enhanced CNN-LSTM* model, which achieves a macro F1-score of *0.9405*, setting a new benchmark for high-accuracy classification in complex Hinglish language data.
- **Optimized Methodology:** A refined methodology detailing an optimal preprocessing pipeline essential for handling code-mixed text, including an analysis of the empirical tradeoffs regarding normalization, noise reduction, and the conversion of non-textual elements like emoticons.

II. LITERATURE SURVEY

A. Traditional Machine Learning for Sentiment Analysis

Classical approaches to sentiment analysis primarily employ supervised *ML* algorithms, such as Naïve Bayes, Logistic Regression, and Support Vector Machines *(SVM)*, which rely on the manual definition and extraction of features.[5] The features are typically derived using models like Bag-of-Words *(BoW)* and Term Frequency–Inverse Document Frequency *(TF-IDF)*. These vectorization techniques convert raw text into a numerical format, where *TF-IDF* assigns weights based on word frequency relative to the corpus, achieving competitive accuracy for tasks on cleaner, higher-resource datasets.[12]

However, traditional *ML* models are constrained by their static nature. They assign a fixed vector representation to each word regardless of its context, rendering them susceptible to misinterpreting polysemy and complex phrases. Furthermore, using all data in a high-dimensional dataset frequently leads to longer computation times and dimensionality issues. In studies evaluating traditional *ML* methods for code-mixed Hinglish, models such as *SVM* and Logistic Regression, even when optimized with uni-bi-trigram *TF-IDF* features, struggled to surpass an F1-score of *0.6800*. The highest performance achieved among this class of models was *0.6907* F1-score, attained by an Ensemble Voting Classifier, demonstrating the severe limitations of static feature modeling when facing the ambiguity inherent in code-mixed text.

B. Evolution to Deep Learning Architectures

Deep Learning (DL) models marked a significant methodological shift by replacing manual feature extraction with multilayer neural network approaches.

- **Recurrent and Convolutional Networks:** Early *DL* models for *NLP* included *Recurrent Neural Networks (RNNs)*, specifically Long Short-Term Memory *(LSTM)* units, which are adept at modeling sequential data and long short-term dependencies critical for sentence-level understanding. *Convolutional Neural Networks (CNNs)* were introduced to identify local patterns, acting as efficient phrase-level feature extractors. Hybrid models combining *CNNs* and *RNNs* have demonstrated superior context-based sentiment results compared to their individual components. One comparative analysis showed a *CNN* model achieving an accuracy of *75.25%* in classifying sentiments, establishing these architectures as a robust baseline.
- **Static Embeddings:** The performance of these *DL* models relies heavily on sophisticated feature representations, such as *Word2Vec* and *Doc2Vec*. These embeddings map words into a low-dimensional vector space, capturing semantic relationships by placing similar words closer together. While an improvement over *TF-IDF*, these remain context-independent, struggling to resolve ambiguities where a word's meaning changes based on the surrounding text (polysemy).

C. Challenges of Code-Mixing and Resource Scarcity

The unique linguistic ecology of India necessitates specialized modeling techniques. The challenge is amplified by data scarcity for many regional languages, leading to poor performance due to a lack of corpora and specialized language resources.

- **Data Strategy:** To address the resource limitations, researchers must rely on limited, specialized corpora, such as the *1000* manually annotated Hindi reviews or the *4000*-review *MABSA* dataset for Malayalam aspect-based sentiment analysis. For resource-poor Dravidian languages like Tamil and Tulu, data augmentation techniques and advanced models are essential to overcome challenges like dataset imbalance and language-specific nuances.
- **Normalization Necessity:** Text normalization is a critical preprocessing step for code-mixed text due to inconsistencies in Romanized spelling. Resources like the hailstorm corpus are specifically created to facilitate the conversion of hybrid language forms into a standardized format. Empirical evidence confirms that implementing robust normalization, particularly for Hindi words, significantly enhances subsequent classifier performance.
- **Cross-Lingual Transfer:** Recognizing that developing massive labeled corpora for every language pair is infeasible, researchers have focused on *Cross-Lingual Sentiment Analysis (CLSA)*. This task leverages classifiers trained in a high-resource language to predict sentiment in a low-resource language. This approach, particularly when using linguistic features like linked *WordNet* senses instead of relying on often inaccurate machine translation systems, has been shown to improve CLSA accuracy by *14% to 15%* for language pairs like Hindi and Marathi. This strategy is fundamental for achieving practical deployment in India's multilingual environment.

D. Contextual Embeddings and Transformer Models

The current trajectory of *NLP* research strongly favors contextual embeddings, which dynamically adjust a word's vector representation based on its position and context within the input text.

Transformer-based models, notably *BERT*, utilize bidirectional attention mechanisms to achieve state-of-the-art contextual understanding. For the Indian context, the adaptation of these models is paramount. Specialized pre-training on code-mixed text, such as in models like HingBERT, has demonstrated superior performance over generic pre-trained models, emphasizing that domain-specific linguistic adaptation is necessary for maximizing accuracy in multilingual tasks.

The most effective strategy involves integrating the robust representation power of Transformers with established sequential and local feature extraction layers. Hybrid models that combine *BERT* embeddings with *CNN* and *LSTM* layers have proven exceptionally powerful for Hinglish sentiment classification. This fusion leverages *BERT's* global semantic grasp, *CNN's* phrase detection capability, and *LSTM's* modeling of long-distance dependencies, achieving high validation accuracies up to *94.1%.*

III. METHODOLOGY: HYBRID DEEP LEARNING APPROACH

The methodology outlined here details the four-part process designed to handle the nuances of Hinglish movie reviews: data acquisition, rigorous preprocessing, feature transformation, and the deployment of a hybrid deep learning architecture.

A. Data Acquisition and Preparation

The study relies on a consolidated, manually annotated movie review corpus consisting of *1000* reviews. This corpus was constructed by combining *250* labeled Hindi movie reviews provided by the *IIT-Bombay* dataset with *750* additional reviews manually crawled and annotated from Hindi movie review websites (e.g., Jagaran.com). The reviews were labeled into two primary sentiment classes: Positive and Negative, maintaining a class balance to avoid bias during training (*500* instances per class). The resultant corpus reflects the real-world occurrence of Hinglish and transliterated Hindi, making it suitable for benchmarking against the core linguistic challenge of this research. The data was split into training, validation, and testing sets to ensure robust evaluation.

B. Preprocessing Pipeline for Code-Mixed Text

A multi-stage, iterative preprocessing pipeline is implemented to maximize the quality of the noisy, user-generated code-mixed data.[3]

- **Initial Cleaning and Consolidation:** This stage involves eliminating null values and discarding irrelevant columns. The raw data, often structured as separated words or tokens, must be consolidated into full review sentences tagged with their corresponding sentiment polarity. Duplicate entries are removed to maintain data integrity.
- **Noise and Symbol Removal:** Specific non-sentimental tokens commonly found in social media text are systematically removed. This includes the elimination of URL links (https), user tags (@ User), hashtag symbols (#), and the Retweet keyword (RT). Punctuation and numerical figures are also addressed in this phase.
- **Stop Word Exclusion:** The process involves the removal of standard English stop words, coupled with the creation and application of a custom Hindi stop word list. This custom list is derived by analyzing short, frequent words in the training data and manually filtering non-stop words to enhance feature significance.
- **Language Normalization:** Crucially, spelling mistakes in both English and transliterated Hindi words are corrected. Normalization of Hindi words using specialized dictionaries is essential, as demonstrated by prior research showing a positive correlation between normalization and improved classifier performance.
- **Emoticon Handling Assessment:** The pipeline included an assessment of converting emoticons into verbal keywords (*e.g.*, using a library like Python's emoji). This experiment revealed that converting emoticons into text and then removing common output words (like 'face') *decreased* classifier performance. Consequently, the optimal configuration for the final model minimizes the loss of emoticons as distinct sentiment features.

C. Feature Transformation and Embedding

The choice of text representation is paramount, determining the model's ability to capture semantic depth.

- **Static Embeddings:** For baseline comparisons, features are generated using the Term Frequency–Inverse Document Frequency (TF-IDF) Vectorizer, which measures the importance of unigrams, bigrams, and trigrams within the corpus.
- **Contextual Embeddings (BERT):** The advanced model utilizes vectors generated by a *Bidirectional Encoder Representations from Transformers (BERT)*

model. Since standard *BERT* lacks optimal performance for code-mixed input, the model employs embeddings derived from a Transformer specifically pre-trained or fine-tuned on Hinglish data. These dynamic embeddings resolve the limitations of static representations by providing context-sensitive vectors, thus effectively modeling the semantics of the mixed language.[4]

D. Model Architectures and Training

1. Traditional ML Baselines

The following traditional models were trained using *TF-IDF* vectors to establish a comparative baseline against the prevailing performance ceiling for code-mixed data [3]: *SVM, KNN*, Decision Trees, Multinomial Naïve Bayes, Logistic Regression, Random Forests, and Ensemble Voting Classifiers.

2. Proposed Hybrid Deep Learning Model: BERT-Enhanced *CNN-LSTM*

The proposed architecture is designed to capture three distinct levels of linguistic information: contextual semantics, local phrasing, and sequential structure.

- **Input Layer:** Receives the context-aware, high-dimensional vector sequences generated by the pre-trained *BERT* model.
- **Convolutional Neural Networks (*CNNs*):** Multiple parallel convolutional layers extract local, fixed-size features, effectively acting as high-level *n-gram* detectors that identify immediate sentiment-bearing phrases in the Hinglish text.
- **Recurrent Neural Networks (*LSTMs*):** Following the *CNN* layers, the features are processed by Long Short-Term Memory units. The *LSTMs* model the temporal or sequential dependencies across the entire review, enabling the network to maintain memory over long sequences and correctly interpret complex structural elements like negation.[4]
- **Training:** The model is fine-tuned using pre-trained Hinglish embeddings and employs strategies like negation handling and feature selection to optimize input prior to classification.

IV. RESULT

A. Performance of Traditional Machine Learning Baselines

The effectiveness of the traditional *ML* models was evaluated using the macro *F1-score*, a crucial metric for evaluating classification models as it represents the harmonic mean of precision and recall, ensuring a balanced measure of accuracy.[23]

Results indicate that Logistic Regression and Random Forest models were the most performant among the individual classical classifiers, with Logistic Regression achieving the best results in six of the comparative experiments. The analysis of *TF-IDF* feature variations showed that using Unigrams or Uni-Bigrams generally yielded superior results compared to using only Trigrams.

The overall best performance achieved by any classical model structure, utilizing the highly optimized Ensemble Voting Classifier with *uni-bi-trigram TF-IDF* features on the normalized dataset, reached a macro *F1-score* of *0.6907.*

Table 1.
Maximum F1-Score for Traditional Models on Code-Mixed Data

Classifier Type	Vectorization	Cleaning Iteration	Macro F1-Score
Ensemble Voting Classifier	TF-IDF (Uni-Bi-Tri-grams)	5 (Normalized)	0.6907
Logistic Regression	TF-IDF (Uni-Bigrams)	5 (Normalized)	0.6797
SVM	TF-IDF (Unigrams)	5 (Normalized)	0.6689

B. Performance of Hybrid Deep Learning Model

The proposed *BERT*-Enhanced *CNN-LSTM* model demonstrated significantly higher performance, achieving state-of-the-art results for the Hinglish movie review corpus.

The overall validation accuracy for the hybrid model was *94.1%.* The macro *F1-score*, reflecting the balance between precision and recall across all classes, was *0.9405.*

Table 2.
Performance Metrics for BERT-Enhanced CNN-LSTM Model

Metric	Value	Interpretation
Validation Accuracy	94.1%	Overall correct classification rate, significantly higher than baseline.
Precision	94.3%	94.3% of positive predictions were correct.
Recall	93.8%	93.8% of actual positive instances were captured.
F1-Score (Macro Avg)	0.9405	High balance between precision and recall.

The detailed breakdown of performance across polarity classes confirms the model's consistent efficacy:

Table 3.
Ensemble Classification Report (BERT-Enhanced CNN-LSTM)

Category	Precision	Recall	F1-Score	Support
Negative	0.95	0.92	0.93	341
Neutral	0.95	0.95	0.95	285
Positive	0.93	0.95	0.94	384
Macro Avg	**0.94**	**0.94**	**0.94**	1010

V. RESULT ANALYSIS

A. Superiority of Contextual Deep Learning

The performance comparison demonstrates a definitive advantage for the contextual deep learning framework. The macro *F1-score* of *0.9405* achieved by the *BERT-*Enhanced *CNN-LSTM* model represents a gain of over *24* percentage points compared to the *0.6907* ceiling of the traditional *ML* ensemble.[3]

This dramatic improvement is primarily attributable to the ability of the Transformer architecture to resolve the semantic ambiguities inherent in code-mixing. Unlike static embeddings *(TF-IDF, Word2Vec)* that assign fixed meanings, BERT generates dynamic vectors that shift based on context. This capability is indispensable for accurately interpreting Hinglish, where the sentiment may rely on the subtle interaction between Hindi and English phrases. Furthermore, the architectural hybridity is essential; the integration of CNN layers ensures the efficient capture of highly localized sentiment-bearing phrases, while the LSTM layers model the overarching narrative and complex logical flow of the review text, which is necessary for high-accuracy contextual classification.[4] The resulting robustness validates the necessity of moving beyond traditional lexical models toward architectures capable of deep linguistic feature learning.

B. Impact of Preprocessing and Feature Engineering

The effectiveness of specialized preprocessing for low-resource and code-mixed languages cannot be overstated. The iterative cleaning process, including the application of customized *stopword* lists and thorough language normalization, established the maximum achievable performance baseline for the traditional models. Specifically, the successful normalization of Romanized Hindi spellings was key, as it ensured that the vector space models could accurately aggregate related tokens, thereby increasing classifier performance.

However, the empirical findings regarding emoticon handling highlight a critical methodological consideration for code-mixed text: not all standard *NLP* preprocessing steps yield positive results. The observation that converting emoticons to keywords negatively impacted performance suggests that, in the context of informal, culturally rich text, emoticons serve as vital, unambiguous sentiment markers. Masking or altering these features effectively removes valuable, low-dimensional signals that the model could utilize directly. This underscores that preprocessing strategies must be tailored and empirically validated for the target linguistic domain, rather than relying on generalized steps suitable for clean, monolingual corpora.

C. Resource Mitigation Strategy Validation

The high performance achieved using a model relying on pre-trained and fine-tuned *BERT* components confirms the viability of transfer learning as a primary resource mitigation strategy for South Asian *NLP* challenges. Given the scarcity of large, fully annotated native corpora for many Indian languages, leveraging powerful architectures pre-trained on vast amounts of general language data and then adapting them to specific code-mixed tasks (*e.g.,* fine-tuning *BERT* for Hinglish) is essential. This approach reduces the dependency on creating massive, perfectly labeled datasets from scratch. The success seen in this study is consistent with other research that demonstrates the outperformance of code-mixed transformer models (like *HingBERT*) over general-purpose language models, affirming that investing in specialized pre-training is crucial for bridging the performance gap in linguistically diverse settings.

VI. FUTURE SCOPE

Future research efforts should focus on expanding the applicability of these robust architectures to address greater linguistic coverage and deeper textual understanding within the Indian cinema domain.

A. Expansion to Multilingual and Cross-Lingual Architectures

While this research successfully addressed Hinglish, the field requires significant expansion into other low-resource Indian languages, including the Dravidian languages (Tamil, Tulu, Kannada) and resource-poor languages like Urdu.[8] Future work must prioritize advancing Cross-Lingual Sentiment Analysis *(CLSA).* This involves developing robust models that utilize linguistic bridging techniques, such as linked *WordNets* or sophisticated lexical resources, rather than unreliable machine translation, to effectively transfer sentiment knowledge from high-resource language corpora to low-resource targets. Furthermore, exploring the optimization of multilingual *BERT (mBERT)* for low-resource Dravidian language tracks remains a promising avenue.

B. Handling Sarcasm and Cultural Context

The detection and analysis of non-literal language, such as sarcasm and humor, remain significant open challenges, particularly because they rely heavily on cultural and contextual incongruity.[25] Sarcasm, which poses computational difficulties due to the lack of nonverbal cues in text [27], can severely distort sentiment polarity. Future models should incorporate explicit sarcasm features into their architecture.[27] Promising directions include utilizing word-emoji embeddings and context-aware analyses to capture the linguistic and emotional cues necessary for sarcasm detection, a process that has previously yielded demonstrable accuracy improvements in sentiment classification.[26]

C. Integration of Multi-Modal Data

Modern movie reviews are often delivered via multi-modal formats, including text accompanied by images, audio, and video commentary.[28] To achieve comprehensive and highly accurate opinion mining, the next generation of frameworks must transition to *Multi-Modal Sentiment Analysis (MSA).*[28] This necessitates developing deep learning architectures that can effectively integrate and fuse diverse data streams (text, audio, visual features).[28] Research into context-aware fusion frameworks that leverage specialized components, such as time-distributed *CNNs* for visual feature extraction and *Bi-LSTM/GRU* layers for textual context, shows that integrating these modalities significantly enhances the detection of nuanced sentiments, humor, and sarcasm within audiovisual content.[30]

REFERENCES

1. R. Patil, G. Galande, V. Kekan, and K. Dange, “Sentiment analysis using support vector machine,” Int J Innov Res Comput Commun Eng., vol. 2, no. 1, 2014.
2. P. Patil and P. Yalagi, “Sentiment analysis levels and techniques: a survey,” Int J Innov Eng Technol., vol. 6, p. 523, 2016.
3. G. Singh, “Sentiment Analysis of Code-Mixed Social Media Text (Hinglish),” 2021. [Online]. Available: arXiv:2102.12149 [cs.CL].
4. V. Kumar, K. K. Pandey, A. K. Rai, and V. Tiwari, “Sentiment Analysis of Hindi-English Code-Mixed Text using Deep Learning Methods,” in 2021 International Conference on Simulation, Automation & Smart Manufacturing (SASM), Mathura, India, Aug. 2021. doi: 10.1109/SASM51857.2021.9841109.
5. N. K. Mishra and B. R. J. B., “Sentiment Analysis of Movie Reviews Using Opinion Words and NLP Toolkit,” in 2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS), Mar. 2024. 31
6. S. Das and T. Singh, “Sentiment Analysis of Hinglish Text using Deep Learning Models,” in 2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, Jan. 2023. doi: 10.1109/Confluence56041.2023.10048879.
7. R. Singh and A. K. Singh, “Sentiment Analysis on Hinglish Reviews using BERT-Enhanced CNN-RNN Hybrid Model,” J. Inf. Technol. Electron. Comput. Eng., vol. 1, no. 1, pp. 60–68, 2024.
8. S. and P. A. N. and S. R. N. and G. V. V. R. and V. K., “Machine Translation for Low-Resource Sentiment Analysis Datasets in Indian Languages,” in Proc. North Am. Chapter Assoc. Comput. Linguistics, 2024, pp. 7689–7698. 32
9. M. R. M. N. R. I., “An Assessment of Machine Learning and Deep Learning Methods for Sentiment Analysis,”

Electronics, vol. 9, no. 3, p. 483, Mar. 2020. doi: 10.3390/electronics9030483.
10. K. J., S. N., N. K., and M. S. C., "Sentiment Analysis using Hybrid Deep Learning Approach," in 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART), Dec. 2021.
11. Singh, B. Kumar, P. Sharma, and K. P. Singh, "Contextual and Static Text Vectorization for Sentiment Analysis: A Comparative Study," 2024. [Online]. Available: https://www.ijraset.com/research-paper/static-and-contextual-text-vectorization-for-sentiment-analysis.
12. S. P. P. and D. S. P. and B. P., "Comparative Analysis of Classification Models for Sentiment Analysis of Movie Reviews with Genre and Aspect Specific Driving Factors," SMU Data Science Review, vol. 5, no. 2, pp. 1–20, 2022.
13. S. A. S. and A. P. P., "A Comparative Study of Deep Learning and Machine Learning Algorithms for Movie Review Sentiment Analysis," in 2018 8th International Conference on Communication Systems and Network Technologies (CSNT), Nov. 2018.
14. H. A. Yacoub, "The role of sarcasm features in enhancing sentiment analysis for text-based content using deep learning," J. Commun. Softw., vol. 20, no. 4, pp. 13–21, 2024. 27
15. A. and P. G., "hinglishNorm: A Corpus for Text Normalization of Hindi-English Code-Mixed Data," 2020. [Online]. Available: arXiv:2010.08974.
16. P. B. and M. N. and D. V. R. and P. A. N., "Hybrid Deep Learning Model for Sarcasm Detection in Indian Indigenous Language Using Word-Emoji Embeddings," Multimed. Tools Appl., vol. 81, pp. 28849–28867, Aug. 2022. 26
17. S. A. G. S. and H. M. P. and S. M. A. A. and M. I. K. M., "MABSA: A Malayalam Aspect-Based Sentiment Analysis Dataset," 2023. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC10415832/.
18. H. R., M. A., and K. A., "Investigating Sentiment Analysis in Resource-Poor Languages: A Systematic Literature Review of Urdu-Based Approaches," Healthcare (Basel), vol. 10, no. 9, p. 1774, Sep. 2022. doi: 10.3390/healthcare10091774.
19. J. H. Park and E. D. R. Santos, "Sentiment Analysis of Code-Mixed Text: A Comprehensive Review," Int. J. Intell. Syst. Appl. Eng., vol. 12, no. 3, pp. 245–256, Feb. 2024.
20. S. Singh, M. Singh, and M. P. S. Bhatia, "Exploring Sentiment Classification Challenges in Dravidian Languages: A Deep Learning Approach for Tamil and Tulu," 2025. [Online]. Available: https://aclanthology.org/2025.dravidianlangtech-1.48.pdf.
21. S. D., "Sentiment Analysis of Hindi Reviews," 2017. [Online]. Available: https://github.com/shubham721/Sentiment-Analysis-On-Hindi-Reviews.
22. J. Abid, H. A. Khan, and H. S. Shah, "Cross-Lingual Sentiment Analysis for Indian using Sense-Based Approach," 2018.
23. M. J. L. and N. J., "Evaluating Transformer Models for Hinglish Named Entity Recognition in the Context of Code-Mixing," 2025. [Online]. Available: https://arxiv.org/html/2509.02514v1.
24. Z. and A. K., "Text Classification for Sentiment Analysis using Vectorization Techniques," Slujst, vol. 1, no. 1, pp. 27–33, 2023.
25. S. S. A. and P. A. N. and S. R. N., "Multimodal Sentiment Analysis: A Survey on Deep Learning Architectures and Applications," 2025. [Online]. Available: https://www.ijcrt.org/papers/IJCRT2501143.pdf.