Sentiment Analysis of Transliterated Hindi and Marathi Using Lexicon-Enriched Transformer Models

Rishikesh Janardan Sutar [1, 2]* and Kamalakar Ravindra Desai[3]
[1]Research Scholar, Department of E&TC Engineering., Department of Technology, Shivaji University, Kolhapur-416004, India
[2]Department of E&TC Engineering, SCTR's PICT, Pune-411043, India
[3]Professor, Department of E&TC Engineering, Bharati Vidyapeeth's College of Engineering, Kolhapur-416013, India
* Corresponding author's Email: rishikeshjsutar@gmail.com

ABSTRACT

This research introduces a structured approach for sentiment analysis in transliterated Hindi and Marathi, two low-resource Indian languages, through a combination of lexicon-driven data generation and enhanced transformer-based modeling. We began by manually curating sentiment lexicons from two authoritative bilingual dictionaries as Oxford Hindi-English and SalaamChaus Marathi-English, selecting 13,231 Hindi and 9,712 Marathi sentiment-bearing words. Each word was manually annotated with a sentiment weight. To address spelling variability in transliterated text, extensive variant forms were generated (176,755 for Hindi, 159,804 for Marathi). Using these, 53,211 Hindi and 30,659 Marathi synthetic sentences were created, with sentence-level sentiment scores derived by averaging the weights of the included sentiment words.

We also created a parallel version of these datasets using publicly available Kaggle sentiment word lists for Hindi and Marathi. Sentence sentiment scores were recalculated based on the Kaggle weights, allowing direct performance comparisons between our manually curated lexicons and an external resource. Additionally, we extracted 11,679 transliterated Hindi comments from YouTube and annotated them with sentiment scores using both our dictionary-based resource and the Kaggle word list, producing two real-world evaluation sets.

To evaluate sentiment classification, we fine-tuned transformer models as MuRIL, XLM-RoBERTa-base, XLM-RoBERTa-large, and IndicBERT, under two experimental setups. In the first, we integrated numerical linguistic features with each transformer model. In the second, we enhanced the models further by incorporating graph-based structural embeddings (via Node2Vec) and applied rank-based feature selection. Results show that our dictionary-based datasets significantly outperformed Kaggle-derived versions for Hindi, mixed Hindi-Marathi, and YouTube comments. For Marathi-only sentences, both resources performed comparably. Notably, incorporating graph embeddings and feature selection further improved accuracy, particularly for Marathi and YouTube datasets. This study highlights the impact of handcrafted lexical resources and structural augmentation in advancing sentiment analysis for underrepresented, transliterated languages.

Keywords: *Sentiment Analysis, Transliterated Languages, Lexicon-based Approach, Transformer Models, Low-resource languages*

1. INTRODUCTION

In the era of digital communication, the proliferation of user generated content on social media platforms has posed new challenges, especially in countries like India, particularly for transliterated text. Transliterated text in Roman script is widely used on social media by bilingual Indian users, who speak but cannot write Hindi or Marathi, making sentiment analysis of such text an emerging research area [1]. Traditional NLP systems, which are optimized for monolingual and grammatically structured languages, often struggle to process such transliterated content. Transliteration is the process of phonetically spelling native language words in a non-native script (often Roman) which adds complexity to sentiment analysis tasks. One of the primary challenges in analyzing transliterated text is the inconsistency in spelling. A single word may have multiple phonetically derived forms, depending on the user, which hampers the effectiveness of standard NLP pipelines. This inconsistency necessitates specialized approaches for sentiment analysis, particularly techniques that can automatically identify the emotional tone behind such irregular text inputs.

While many researchers have explored sentiment analysis in native scripts or in standardized Roman transliteration formats, there remains a significant gap in the analysis of transliterated Hindi and Marathi using supervised machine learning models on richly annotated datasets that include spelling variations. Additionally, sentence-level sentiment analysis can be improved by expanding sentiment dictionaries to include newly observed or context-specific words [2]. This research addresses the gap by first extracting sentiment-bearing words from standard bilingual dictionaries [3-4] to form a core vocabulary. It then constructs a novel, transliteration-aware sentiment dictionary by generating spelling variants. This expanded resource is evaluated using supervised learning with modified transformer-based classifiers, aimed at overcoming the linguistic and structural challenges of transliterated Indian language sentiment analysis.

2. LITERATURE SURVEY

Ansari and Govilkar [5] performed sentiment analysis on 1200 Hindi and 300 Marathi transliterated social media posts using KNN, Naïve Bayes, SVM, and ontology-based methods, achieving up to 80% accuracy for Hindi. They emphasized the need for enhanced POS tagging and enriched lexical resources. Srinivasan and Subalalitha [6] analyzed a Tamil-English code-mixed dataset (15,744 sentences) using RF, SVM, LR, and XGBoost. Applying SMOTE/ADASYN for class balance, they achieved an F1-score of 0.81. Their future work highlighted handling spelling variations. Khare and Khan [7] surveyed ML and DL techniques for Hindi sentiment analysis, pointing out challenges from dialects and code-mixing. They proposed the development of dialect-specific tools and annotated corpora. Pandey and Govilkar [8] compared sentiment analysis across Hindi, Bengali, and Punjabi, with their modified HSWN reaching ~80% accuracy. The study stressed hybrid approaches and negation handling.

Alam et al. [9] reviewed 91 studies (2010–2024), spanning ML, DL, and transformer-based sentiment models. Future trends emphasized multimodal emotion detection, interpretability, and ethical AI. Sharma et al. [10] utilized Hindi SentiWordNet with N-Gram and synset replacement techniques on movie reviews, effectively capturing negation. They called for expanding lexicons and standardizing datasets. Horvat et al. [11] developed a hybrid NLP model using ANEW and NRC lexicons for Croatian crisis-related texts, highlighting emotional variation. Future work included multilingual and granular emotion modeling. Sidhu et al. [12] reviewed Hindi SA using methods like SVM and MT, showing ~80% accuracy and stressing the need for annotated corpora, stemmers, and DL tools. Chanda et al. [13] employed mBERT on Dravidian-CodeMix 2021 datasets, achieving ~0.61 accuracy. They emphasized improving tagging and context modeling. Mulatkar and Bhojane [14] proposed a rule-based Hindi SA system using WordNet and SentiWordNet, advocating integration of statistical and semantic techniques. Shekhar et al. [15] implemented an LSTM model optimized via artificial immune systems for Hindi-English code-mixed texts. They claimed improved ambiguity resolution and suggested multilingual generalization. Kumar et al. [16] provided a broad review from lexicon-based to transformer models like BERT and GPT, urging work on cross-lingual sentiment analysis and explainable AI. Rani and Kumar [17] focused on CNN-based SA for Hindi, noting its effectiveness in feature extraction. They proposed CNN-LSTM hybrids as future scope. Ahamad and Mishra [18] introduced the ESIHE_AML model (CNN + Bi-LSTM), surpassing 90% accuracy on Twitter and Amazon datasets, suggesting further hybrid model development.

Sharma et al. [19] discussed sarcasm detection, domain adaptation, and multilingual modeling challenges, recommending hybrid DL models with ethical considerations. Sharma and Lakhwani [20] reviewed 34 cross-domain SA papers, pointing to domain shift issues and advocating dynamic feature augmentation. Sazan et al. [21] used CNN-BiLSTM with BERT/TF-IDF for Bangla mental health data, achieving an 84% F1-score. They emphasized the creation of multilingual tools and public datasets. Yadav et al. [22] conducted dictionary-based SA on Hindi news using polarity lists, identifying polysemy issues and proposing better context handling. Shelke and Deshmukh [23] reviewed SA across Indian languages using HSWN. Their Hindi system HOMS achieved 91.4% accuracy, highlighting the need for ontology-based methods and support for low-resource languages. Pawar and Mali [24] conducted SA on Marathi using SVM and NB, noting a lack of resources and tools. Gupta and Ansari [25] examined Hindi SA in blogs using ML, underlining the need for annotated corpora and tailored tools. Bhoir et al. [26] applied lexicon-based SA to Marathi tweets, recommending expansion via ML techniques. Lomte et al. [27] reviewed Marathi SA for speech and text, with 74–97% accuracy using MSVM, ANN, and HMM. They advocated for better datasets and speech processing. Thorat and Guide [28] reviewed Hindi SA using CNN, DNN, and RNN, emphasizing the development of tools for code-mixed and morphological analysis. Ranjan and Poddar [29] developed an abuse detection system for Moj data using dictionary-free spell correction, suggesting broader Indic language applications. Liu et al. [30] showed transliteration improving Hindi-Urdu alignment but not always benefiting downstream tasks. Eusha et al. [31] analyzed Tamil and Tulu code-mixed text with transformers, achieving F1-scores of 0.23 and 0.58, respectively. They underscored the need for better datasets and models for low-resource languages.

As evident from this comprehensive survey, a recurring theme across the reviewed works is the critical need for enriched datasets tailored to transliterated and low-resource languages. Many studies emphasize the challenges posed by transliteration, spelling variations, and the absence of standardized sentiment lexicons, particularly for Indian languages like Hindi [1,12,25,28] and Marathi [2,23]. Accordingly, several researchers propose the development of well-annotated, language-specific resources as a future direction to enhance sentiment classification performance. In alignment with these insights, the present work focuses on the creation of a manually curated sentiment word dataset with associated sentiment weights for transliterated Hindi and Marathi. These enriched lexicons serve as the foundation for generating sentence-level datasets with computed sentiment scores, ultimately enabling robust training and evaluation of sentiment analysis models in low-resource, code-mixed contexts.

3. BLOCK SCHEMATIC OF PROPOSED WORK

The proposed sentiment analysis framework targets transliterated Hindi and Marathi by first extracting sentiment-bearing words from the Oxford Hindi-English and SalaamChaus Marathi-English dictionaries. Each word is manually assigned a sentiment score based on contextual polarity. To address transliteration inconsistencies, extensive spelling variants are generated for each word, improving robustness against noisy real-world inputs. Using these enriched lexicons, synthetic sentences containing at least two sentiment words are constructed. In parallel, sentiment word lists from Kaggle [32] are also used to generate comparable sentence datasets.

For each sentence, an average sentiment score is calculated using both custom and Kaggle lexicons. Additionally, 11,679 real-world transliterated Hindi YouTube comments are curated, with sentiment scores computed using both lexicon sets. This results in comprehensive datasets of Hindi, Marathi, and YouTube sentences with corresponding sentiment labels derived from two lexicon sources.

Sentiment classification is performed using modified MuRIL, XLM-RoBERTa-base, XLM-RoBERTa-large, and IndicBERT models under two settings: (1) using semantic embeddings with handcrafted numeric features, and (2) enhancing these with graph-based structural embeddings and SelectKBest feature selection. Model performance is evaluated using accuracy, with comparisons drawn across datasets and configurations to assess the contribution of graph embeddings and feature selection.

The proposed framework for sentiment analysis in transliterated Hindi and Marathi languages is as illustrated in the block schematic shown in above Figure 1.

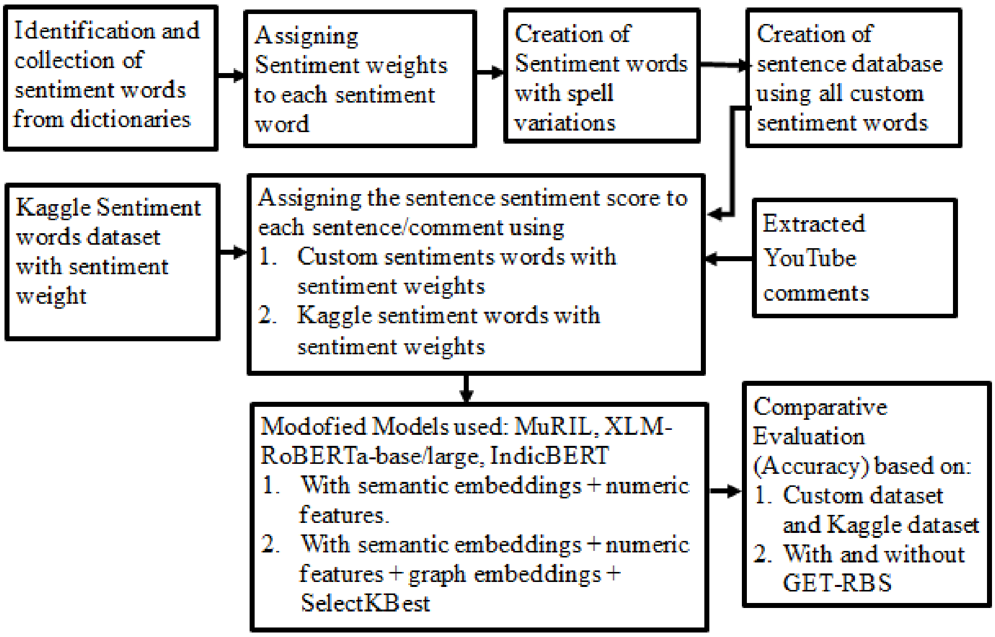
Identification and collection of sentiment words from dictionaries
Assigning Sentiment weights to each sentiment word
Creation of Sentiment words with spell variations
Creation of sentence database using all custom sentiment words
Kaggle Sentiment words dataset with sentiment weight
Assigning the sentence sentiment score to each sentence/comment using
1. Custom sentiments words with sentiment weights
2. Kaggle sentiment words with sentiment weights
Extracted YouTube comments
Modofied Models used: MuRIL, XLM-RoBERTa-base/large, IndicBERT
1. With semantic embeddings + numeric features.
2. With semantic embeddings + numeric features + graph embeddings + SelectKBest
Comparative Evaluation (Accuracy) based on:
1. Custom dataset and Kaggle dataset
2. With and without GET-RBS

Figure 1. Block schematic of proposed work

4. METHODOLOGY

This section describes the construction of sentiment-annotated datasets, feature extraction pipelines, and transformer-based classification architectures. Our approach combines lexicon-driven sentiment scoring, synthetic data generation, real-world text annotation, and multimodal feature integration with transformer models.

4.1 Lexicon-Based Sentiment Annotation

Let W_{Hi} and W_{Mr} be the sets of sentiment-bearing words manually extracted from the Oxford Hindi-English and SalaamChaus Marathi-English dictionaries, respectively. For each word w_i∈W, a **word sentiment score (WSS)** is assigned as:

$$s(w_i) \in \{-3, -2, -1, 0, +1, +2, +3\} \quad (1)$$

To address transliteration variance, each word w_i is expanded into a set of k possible spelling variants:

$$Var(w_i)=\{w_i^{(1)}, w_i^{(2)}, w_i^{(3)}, \ldots, w_i^{(k)}\} \quad (2)$$

Spell varied sample for Marathi and Hindi word is as below:

praanaahun priya: pranahun priya2m+3, pranahun priy2m+3, pranahoon priya2m+3, pranahoon priy2m+3, pranahoon priyaa2m+3, pranahun priyaa2m+3, praanahun priya2m+3, praanahoon priya2m+3, praanahun priyaa2m+3, praanahoon priyaa2m+3, praanahoon priy2m+3, praanahun priy2m+3, pranahun preya2m+3, pranahoon preya2m+3, pranahun priyaa2m+3, praanhun priya2m+3, praanhun priyaa2m+3, praanhun priy2m+3, pranahoon preyaa2m+3, pranahun preyaa2m+3, praanhun preya2m+3, praanahoon preya2m+3, pranahoon prey2m+3, pranahun prey2m+3, pranahoon priyaa2m+3, praanaahun priya2m+3, prraannaahun priya2m+3, prraanaahun priya2m+3, prannaahun priya2m+3, praannaahun priya2m+3, prrannaahun priya2m+3, prranaahun priya2m+3, praanaahun priya2m+3, pranaahun priya2m+3.

In “pranahun priya2m+3”, “pranahun Priya” is the sentiment word, “2” is for the number of sub-words from main sentiment word, “m” is for Marathi language, “+3” is the sentiment weight assigned for the sentiment word.

akelaapan : akelapan1h-2, akelapanh1h-2, akelapann1h-2, akeilapan1h-2, akelapaan1h-2, akelapun1h-2, akelapn1h-2, akailapan1h-2, akailapann1h-2, akelaphan1h-2, akelaphann1h-2, akelappan1h-2, akeilapaan1h-2, akaylapan1h-2, akelappann1h-2, akelaphaan1h-2, akaylapaan1h-2, akelapahn1h-2, akailapn1h-2, akaylapanh1h-2, akelapna1h-2, akelaphaann1h-2, akailapaanh1h-2, akailapahn1h-2, akelaphahn1h-2, akailappan1h-2, akelappaan1h-2, akailappaan1h-2, akailaphan1h-2, akelapanh1h-2, akaylapn1h-2, akailapnah1h-2, akailaphahn1h-2, akelapahn1h-2, akelaapan1h-2, akellapann1h-2, akellapan1h-2, akellaapan1h-2, akellaapann1h-2, akelaapann1h-2.

In "akelapan1h-2", "akelapan" is the sentiment word, "1" is the number of sub-words from sentiment word, "h" is for Hindi language, "-2" is for sentiment weight assigned for sentiment word.

Let S={x_1,x_2,...,x_N} be the set of synthesized sentences formed using these variants. For each entence x_j , containing n_j matched sentiment words {$w_j^{(1)}$, $w_j^{(2)}$,...,$w_j^{(n_j)}$}, the **sentence sentiment score** (SSS) is computed as:

$$SSS_{before\_roundingup}(x_j)=\frac{1}{n_j}\sum_{i=1}^{n_j} s(w_j^{(i)}) \qquad (3)$$

Rounding up of SSS is given as,

$$\text{SSS}(x_j)=\text{ceil}(SSS_{before\_roundingup}\,(x_j)) \qquad (4)$$

A few samples of sentences along with sentiment scores are shown in Table 1.

Table 1. Few samples of sentiment sentences with calculated sentence sentiment score (SSS)

Sentence preprocessed	Actual average sentence sentiment score	No. of Hindi words	No. of Marathi words	Rounded sentence sentiment score
akaaj vyakti ka jeevan hamesha akad aur ahankaar se bhara hota hai	-1.666666667	3	0	-2
khushnumaa aur khushmijaaj thi wo jo har jagah apna rang daal deti thi	2.333333333	3	0	3
abhishek karna ek pavitra karya hai jo devtaon ke liye kiya jata hai	1.5	2	0	2
ghamendee vartan karanaaryaalaa ghamendkhor mhanatat	-2	0	2	-2
taalbaddha sangeet ani taalmel sundar vatato	1.666666667	0	3	2
shaantataapriya lok shaaleen astat	2.5	0	2	3

This process generated 53,211 Hindi and 30,659 Marathi sentences. Each sentence is labeled by rounding the score to the nearest integer within [−3, +3], forming a 7-class classification task.

4.2 Dataset Variants and Real-World Extension

We replicated the scoring process using a publicly available Kaggle sentiment word list W_{Kaggle}, yielding alternate sentiment labels for the same sentences.

Furthermore, we collected 11,679 transliterated Hindi YouTube comments Y={$y_1, y_2, y_3, \dots, y_{11679}$}, and computed two versions of sentence scores for each comment:

- $SSS_{dict}(y_i)$ using our lexicon,
- $SSS_{Kaggle}(y_i)$ using Kaggle weights.

4.3 Feature Extraction

Each sentence or comment x_i is encoded into a feature vector composed of the following components:

(a) Semantic Embedding $\boldsymbol{z_i \in R^d}$

Generated by extracting the [CLS] token embedding from a pretrained transformer model f_θ, such as MuRIL or XLM-R:

$z_i = f_\theta(x_i)_{[CLS\}}$ (5)

(b) Numeric Feature Vector $\boldsymbol{n_i \in R^5}$

Includes sentence-level linguistic features as average sentiment score, count of sentiment words in Hindi, Marathi, English, and total number of words.

(c) Graph Embedding $\boldsymbol{g_i \in R^{64}}$

Sentences are treated as nodes in an undirected graph G=(V, E), where each node is linked to adjacent sentences to simulate local transitions. We apply **Node2Vec** with biased random walks and Skip-gram optimization to generate structural embeddings g_i.

4.4 Feature Fusion and Dimensionality Reduction

We concatenate all features:

$h_i = [z_i \| n_i \| g_i] \in R^{d+5+64}$ (6)

For models using graph embeddings, we apply **rank-based feature selection** to reduce noise:

$h_i^{(sel)}$=SelectKBest(h_i,k) (7)

4.5 Classification Models

Each transformed feature vector $h_i^{(sel)}$ is passed to a feedforward neural classifier:

$\hat{y}_i$=arg max Softmax($f_\phi(h_i^{(sel)})$) (8)

where f_ϕ is a multilayer perceptron with batch normalization and dropout regularization. All models were trained using the CrossEntropy loss function:

L=$-\sum_{i=1}^{N} \log\,(y_i \,|h_i^{(sel)})$ (9)

4.6 Experimental Configurations

We evaluated two cases:

- **Case X**: semantic embeddings + numeric features.
- **Case Y**: semantic embeddings + numeric features + graph embeddings + SelectKBest.

Each configuration was tested on all datasets using MuRIL, XLM-RoBERTa-base, XLM-RoBERTa-large, and IndicBERT, with consistent training parameters (learning rate = 2×10^{-5}, batch size = 16–32, epochs = 5–7).

4.7 Evaluation Metrics

Model performance was assessed using accuracy.

Accuracy: Measures the proportion of correctly classified sentences out of the total sentences as per Eq. (10).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (10)$$

where, TP is True Positive, FP is False Positive, TN is True Negative, and FN is False Negative.

5. RESULTS

The results are reported in terms of accuracy across different configurations, comparing the performance of the custom lexicon with the publicly available Kaggle sentiment lexicon. Evaluations were conducted using multiple transformer models, and the impact of incorporating Graph Embeddings and Rank-Based Selection (GET+RBS) was also analyzed. Accuracy values corresponding to each lexicon and model configuration are summarized in Table 2.

Spell varied custom lexicon outperformed for Hindi and mixed Hindi-Marathi sentences along with YouTube comments as shown in Figure 2. Use of GET+RBS improved accuracy for Marathi sentences and YouTube comments as shown in Figure 3.

Table 2: Comparative Accuracy (%) Across Datasets and Experimental Settings

Dataset	Transformer model	Case X (without GET & RBS)		Case Y (with GET & RBS)	
		With custom Lexicon	With Kaggle Lexicon	With custom Lexicon	With Kaggle Lexicon
Hindi	MuRIL	**0.97**	0.94	0.96	0.96
	XLM-RoBERTa Base	**0.96**	0.93	0.95	0.96
	XLM-RoBERTa Large	**0.95**	0.84	0.96	0.93
	IndicBERT	**0.95**	0.92	0.94	0.92
Marathi	MuRIL	0.88	0.90	**0.94**	**0.95**
	XLM-RoBERTa Base	0.87	0.89	**0.95**	**0.96**
	XLM-RoBERTa Large	0.87	0.85	**0.95**	**0.96**
	IndicBERT	0.88	0.90	**0.92**	**0.95**
Hindi + Marathi (Combined)	MuRIL	**0.96**	0.94	0.96	0.91
	XLM-RoBERTa Base	**0.95**	0.93	0.97	0.95
	XLM-RoBERTa Large	**0.91**	0.69	0.98	0.96
	IndicBERT	**0.96**	0.95	0.95	0.93
YouTube Comments (Real-world)	MuRIL	**0.80**	0.61	**0.94**	**0.91**
	XLM-RoBERTa Base	**0.73**	0.62	**0.93**	**0.89**
	XLM-RoBERTa Large	**0.74**	0.63	**0.95**	**0.90**
	IndicBERT	**0.76**	0.60	**0.94**	**0.89**

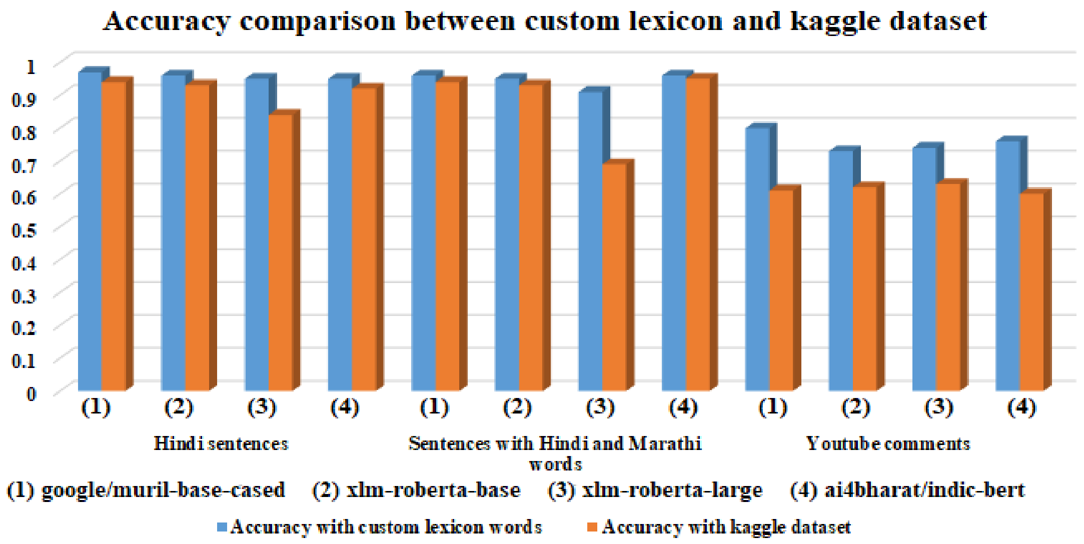

Figure 2. Accuracy comparison between custom lexicon and Kaggle dataset

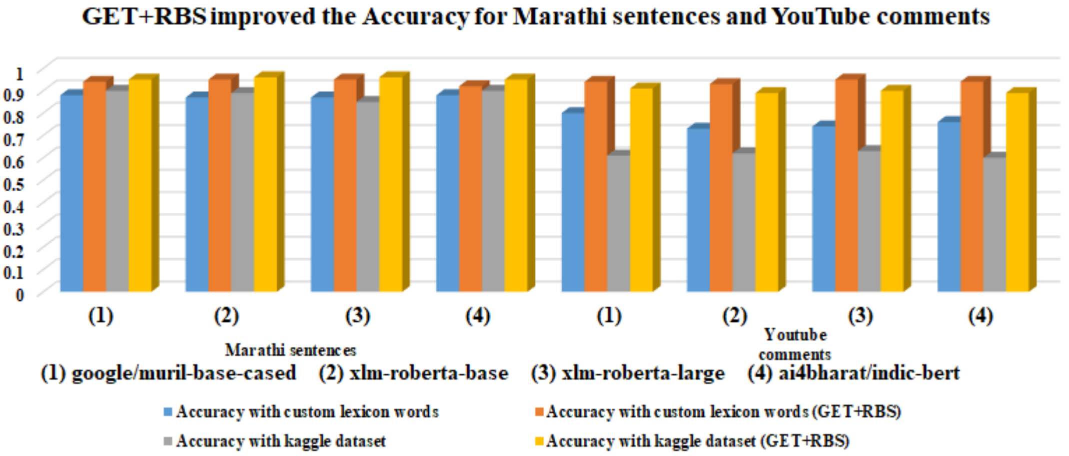

Figure 3. Accuracy comparison with GET+RBS in Marathi sentences and YouTube comments

6. DISCUSSION

The experimental results underscore the effectiveness of the custom-created sentiment lexicons and sentence datasets in improving classification performance across transliterated Hindi and Marathi texts. Models trained using the manually curated, spell-varied datasets consistently outperformed those trained on versions derived from publicly available Kaggle sentiment word lists. This performance gap was particularly notable in the Hindi and mixed Hindi-Marathi sentence datasets, as well as in the real-world YouTube comment dataset. The improvement can be attributed to the significantly larger lexical coverage and orthographic diversity in the custom dataset, which allowed models to better capture sentiment expressions prevalent in noisy, user-generated transliterated content. In contrast, the Kaggle-based lexicons lacked spelling variants and domain-specific expressions, leading to reduced accuracy and generalization.

Further performance gains were observed when graph-based structural embeddings (via Node2Vec) and rank-based feature selection (SelectKBest) were introduced in the second experimental setting. This configuration, referred to as GET+RBS, yielded the most benefit on the Marathi and YouTube datasets, where linguistic structure and word co-occurrence patterns are more critical due to sparse sentiment cues or limited vocabulary overlap. The GET+RBS strategy enhanced the model's ability to focus on informative patterns while reducing the impact of redundant features. Interestingly, while both experimental setups performed comparably for Hindi datasets, the addition of structural and rank-based signals improved consistency and robustness for low-resource or informal language scenarios. Across all the MuRIL, XLM-RoBERTa (base and large), and IndicBERT models, the results confirm that the quality and contextual relevance of the training data played an equal and more significant role in model performance than the choice of transformer architecture alone.

7. CONCLUSION AND FUTURE SCOPE

This study presented a comprehensive framework for sentiment analysis of transliterated Hindi and Marathi texts, leveraging manually curated sentiment lexicons, synthetically generated sentence datasets, and real-world YouTube comments. A total of over 22,000 sentiment-bearing words were extracted from authoritative bilingual dictionaries, with sentiment weights assigned manually. Extensive spelling variation was introduced to simulate the irregularities of transliterated text commonly found in social media. These enriched word sets were used to generate large-scale sentence datasets with sentence-level sentiment scores, forming the basis for robust supervised classification tasks.

To assess the effectiveness of the proposed resources, transformer-based models (MuRIL, XLM-RoBERTa-base/large, and IndicBERT) were trained using two experimental setups, one relying solely on textual and numeric features, and another incorporating graph-based structural embeddings and rank-based feature selection. Across multiple datasets, the models trained on the manually created, spelling-variant-rich lexicons outperformed those using publicly available Kaggle word lists, particularly on real-world YouTube comment data. The integration of graph embeddings and feature selection further improved performance in select cases, highlighting the importance of structural and statistical cues in low-resource, code-mixed language settings.

Future work may address linguistic challenges such as sarcasm detection and flexible word order, which remain difficult in code-mixed, transliterated languages. Incorporating syntactic parsing or attention-based mechanisms could improve sentiment classification. Additionally, the proposed lexicon creation and spelling-variant generation approach can be extended to other low-resource Indian languages, enabling broader applicability and improved sentiment analysis in similar settings.

8. REFERENCES

[1] M. Thomas, and C. Latha, "Sentimental analysis of transliterated text in Malayalam using recurrent neural networks", *Journal of Ambient Intelligence and Humanized Computing,* 2020, doi: 10.1007/s12652-020-02305-3.

[2] S. Deshmukh, N. Patil, S. Rotiwar, and J. Nunes, "Sentiment Analysis of Marathi Language", *International Journal of Research Publications in Engineering and Technology [IJRPET],* Vol. 3, Issue 6, 2017, pp. 93-97.

[3] Oxford Hindi-English Dictionary

[4] Salaamchaus's Marathi-English Dictionary

[5] A. Ansari, and S. Govilkar, "Sentiment Analysis of Mixed code for the Transliterated Hindi and Marathi Texts", *International Journal on Natural Language Computing (IJNLC)*, Vol. 7, No.2, 2018, doi: 10.5121/ijnlc.2018.7202.

[6] R. Srinivasan, and C. Subalalitha, "Sentimental analysis from imbalanced code-mixed data using machine learning approaches", *Distributed and Parallel Databases,* doi: 10.1007/s10619-021-07331-4.

[7] B. Khare, and I. Khan," Machine Learning Approaches for Sentiment Analysis in Hindi Text: A Comprehensive Survey", *International Journal of Innovative Research in Computer Science & Technology (IJIRCST),* Vol. 12, Special Issue-1, 2024, doi: 10.55524/CSISTW.2024.12.1.62.

[8] P. Pandey, and S. Govilkar, "A survey of Sentiment Classification techniques used for Indian regional languages", *International Journal on Computational Science & Applications (IJCSA),* Vol.5,

No.2, 2015, pp. 13-26, doi:10.5121/ijcsa.2015.5202.

[9] S. Alam, S. Mrida, and A. Rahman, "Sentiment Analysis in social media: How data science impacts public opinion knowledge integrates Natural Language Processing (NLP) with Artificial Intelligence (AI)", American *Journal of Scholarly Research and Innovation*, 4(01), 2025, pp. 63-100, doi: 10.63125/r3sq6p80.

[10] S. Sharma, S. Bharti, and R. Goel, "A Frame Study on Sentiment Analysis of Hindi Language Using Machine Learning", *International Journal of Trend in Scientific Research and Development*, Vol. 2, 1603-1607, doi: 10.31142/ijtsrd14397.

[11] M. Horvat, G. Gledec, and F. Leontić, "Hybrid Natural Language Processing Model for Sentiment Analysis during Natural Crisis", *Electronics 2024*, 13, 1991, doi: 10.3390/electronics13101991.

[12] S. Sidhu, S. Khurana, M. Kumar, P. Singh, and S. Bamber, "Sentiment analysis of Hindi language text: a critical review", *Multimedia Tools and Applications*, 2023, doi: 10.1007/s11042-023-17537-6.

[13] S. Chanda, A. Mishra, and S. Pal, "Sentiment analysis of code-mixed Dravidian languages leveraging pretrained model and word-level language tag," *Natural Language Processing*, Vol. 31, No. 2, pp. 477–499, 2025. doi:10.1017/nlp.2024.30.

[14] S. Mulatkar, and V. Bhojane, "Sentiment Classification in Hindi", *IOSR Journal of Computer Engineering (IOSR-JCE)*, Vol. 17, Issue 4, 2015, PP 100-102, doi: 10.9790/0661-1741100102.

[15] S. Shekhar, D. Sharma, D. Agarwal, and Y. Pathak, "Artificial Immune Systems-Based Classification Model for Code-Mixed Social Media Data", *IRBM*, (2020), doi: 10.1016/j.irbm.2020.07.004.

[16] M. Kumar, L. Khan, and H-T Chang, "Evolving techniques in sentiment analysis: a comprehensive review", *PeerJ Comput. Sci. 11: e2592*, 2025, doi:10.7717/peerj-cs.2592

[17] S. Rani, and P. Kumar, "Deep Learning Based Sentiment Analysis Using Convolution Neural Network", *Arabian Journal for Science and Engineering*, 2019, doi: 10.1007/s13369-018-3500-z.

[18] R. Ahamad, and K. Mishra, "Exploring sentiment analysis in handwritten and E-text documents using advanced machine learning techniques: a novel approach", *J Big Data 12*, 2025, doi: 10.1186/s40537-025-01064-2.

[19] N. Sharma, S. Ali, and A. Kabir, "A review of sentiment analysis: tasks, applications, and deep learning techniques", *Int J Data Sci Anal* 19, 351–388, 2025, doi: 10.1007/s41060-024-00594-x.

[20] R. Sharma, and K. Lakhwani, "A Systematic Literature Review on Cross Domain Sentiment Analysis Techniques: PRISMA Approach", *Annals of Emerging Technologies in Computing (AETiC)*, Vol. 8, No. 4, 2024, doi: 10.33166/AETiC.2024.04.002.

[21] S. Sazan, M. Miraz, and M. Rahman, "Enhancing Depressive Post Detection in Bangla: A Comparative Study of TF-IDF, BERT and FastText Embeddings", *Annals of Emerging Technologies in Computing (AETiC)*, Vol. 8, No. 3, 2024, doi: 10.33166/AETiC.2024.03.003.

[22] O. Yadav, R. Patel, Y. Shah, and S. Talim, "Sentiment Analysis on Hindi News Articles", *International Research Journal of Engineering and Technology (IRJET)*, Vol. 07 Issue: 05, 2020.

[23] M. Shelke, and S. Deshmukh, "Recent Advances in Sentiment Analysis of Indian Languages", *International Journal of Future Generation Communication and Networking*, Vol. 13, No. 4, (2020), pp. 1656–1675.

[24] S. Pawar, and S. Mali, "Sentiment Analysis in Marathi Language", *International Journal on Recent*

and Innovation Trends in Computing and Communication, 2017, Vol. 5, Issue: 8, pp. 21-25.

[25] S. Gupta, and G. Ansari, "Sentiment Analysis in Hindi Language: A Survey", *International Journal of Modern Trends in Engineering and Research (IJMTER),* Vol. 01, Issue 05, 2014, pp. 82-88.

[26] N. Bhoir, A. Das, M. Jakate, S. Lavangare, and D. Kadam, "A Study on Sentiment Analysis of Twitter Data for Devnagari Languages", *International Research Journal of Engineering and Technology (IRJET),* Vol. 08 Issue: 10, 2021.

[27] V. Lomte, P. Jadhav, O. Kalshetti, S. Deshmukh, and A. Jadhav, "Survey on Sentiment Analysis of Marathi Speech and Script", *International Research Journal of Engineering and Technology (IRJET),* Vol. 08 Issue: 12, 2021, pp. 876-893.

[28] M. Thorat, and N. Guide, "Review Paper on Sentiment Analysis for Hindi Language", *Grenze International Journal of Engineering and Technology, Jan Issue, Grenze Scientific Society,* 2022, Grenze ID: 01. GIJET.8.1.74.

[29] E. Ranjan, and N. Poddar, "Multilingual Abusiveness Identification on Code-Mixed Social Media Text", *arXiv:2204.01848v1 [cs.CL],* 2022.

[30] Y. Liu, M. Wang, A. Kargaran, A. Imani, O. Xhelili, H. Ye, C. Ma, F. Yvon, and H. Schütze, "How Transliterations Improve Crosslingual Alignment", *arXiv:2409.17326,* 2024.

[31] A. Eusha, S. Farsi, A. Hossain, S. Ahsan, and M. Hoque, "Sentiment Analysis using Transformer-Based Models in Code-Mixed and Transliterated Tamil and Tulu", *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages,*2024, pp. 205–211.

[32] Kaggle dataset link: https://www.kaggle.com/datasets/warcoder/transliteration-dataset-21-indic-languages