

Emoji Aware Multilingual Sentiment Analysis for Marathi and Hindi

Swapnali Purushottam Kulthe¹, Shubhangi S. Mandwale², Prof. Abhijeet Khadke³

¹*M. tech in CSE, Department of Computer Science and Engineering, Shreeyash College of Engineering and Technology, Chh. Sambhajinagar*

²*Professor Department of Computer Science and Engineering, Shreeyash College of Engineering and Technology, Chh. Sambhajinagar*

³*HOD Department of Computer Science and Engineering, Shreeyash College of Engineering and Technology, Chh. Sambhajinagar*

Abstract—This paper presents an emoji-aware multilingual sentiment-analysis system for Marathi and Hindi short text. The system preserves Unicode pictographs, enriches them with descriptive tokens, and compares three modelling families under a common data split and interface: TF-IDF with a linear support-vector machine, fastText representations with logistic regression, and the MuRIL Transformer. On the Marathi held-out set, the models obtained accuracy and macro-F1 scores of 0.61/0.61, 0.65/0.65, and 0.804/0.803, respectively. On the Hindi IITP data, MuRIL achieved 0.6419 accuracy and 0.5507 macro-F1, exceeding TF-IDF and fastText. The results show that contextual pre-training is most effective overall, while classical and subword baselines remain valuable when speed, storage, and interpretability matter. A Streamlit dashboard supports training, batch prediction, confidence inspection, confusion matrices, and artifact-isolated model comparison. The study also identifies the limits imposed by small samples, domain shift, class imbalance, code-mixing, and context-dependent emoji meaning.

Index Terms—sentiment analysis, emoji, Marathi, Hindi, MuRIL, fastText, TF-IDF, multilingual NLP

I. INTRODUCTION

Sentiment analysis converts subjective text into polarity labels such as positive, negative, and neutral. Its practical value extends to customer feedback, public-policy monitoring, healthcare, education, tourism, and social-media analytics. Early lexicon and frequency-based systems offered efficient baselines, but they struggled with negation, implicit sentiment,

sarcasm, and domain-specific meaning [1], [2]. Transformer models improved contextual representation through self-attention, yet their computational cost and dependence on representative training data remain significant [3], [4].

The problem is harder for Marathi and Hindi because annotated corpora are relatively limited, morphology is rich, spelling varies, and users frequently mix languages and scripts. Informal messages also use emoji as sentiment indicators, intensifiers, mitigators, or reversers. Removing symbols during preprocessing can therefore discard essential evidence, particularly in short utterances [5], [6].

This work addresses that gap with a unified comparison framework. It retains emoji, normalizes multilingual text without destroying script information, trains three independent model families, and exposes both predictions and evaluation evidence through one interface. The objective is not only to maximize accuracy, but also to understand the trade-off among contextual quality, training cost, storage, and reproducibility.

II. RELATED WORK

Classical sentiment systems represent text through Bag-of-Words, n-grams, or TF-IDF and apply classifiers such as Naive Bayes, logistic regression, or SVM. TF-IDF with a linear SVM is particularly effective in high-dimensional sparse spaces and remains a transparent baseline for limited data [1], [7].

It cannot, however, distinguish meanings that depend on word order or wider context.

Distributed word representations improved generalization by placing semantically related words near one another. fastText extends this idea with character n-grams, which helps with inflection, spelling variation, and unseen word forms in morphologically rich languages [8]. Sentence-level averaging remains shallow because it loses much of the sequence structure. Transformers address this limitation using self-attention [3]. BERT established contextual fine-tuning, while multilingual models such as XLM-R and IndicBERT enabled cross-lingual transfer [4], [9], [10]. MuRIL further targets Indian languages and transliterated pairs, making it suitable for Hindi, Marathi, and code-mixed inputs [11].

Dataset development has also advanced Indian-language research. L3CubeMahaSent and MahaSent-MD provide three-class Marathi data across social and multi-domain settings [12], [13]. Hindi benchmarks include annotated review resources and BHAAV-style emotion or sentiment collections [14], [15]. Emoji-aware studies show that pictographs offer useful weak supervision and polarity cues, but their meaning can vary across culture, language, and discourse context [5], [6].

Table 1 Summary of representative related work

Study	Approach or resource	Contribution	Limitation
Salton and Buckley [1]	TF-IDF	Efficient, interpretable term weighting	No contextual semantics
Cortes and Vapnik [7]	Linear SVM	Strong sparse-text decision boundary	Probabilities need calibration
Bojanowski et al. [8]	fastText	Subword handling for morphology and OOV words	Averaging loses word order

Study	Approach or resource	Contribution	Limitation
Vaswani et al. [3]	Transformer	Long-range contextual representation	Compute and data intensive
Khanuja et al. [11]	MuRIL	Indic scripts and transliteration	Large model and slower training
Kulkarni et al. [12]	L3CubeMahaSent	Large Marathi three-class corpus	Twitter-domain bias
Pingle et al. [13]	MahaSent-MD	Cross-domain Marathi evaluation	Domain shift remains difficult
Kralj Novak et al. [5]	Emoji Sentiment Ranking	Multilingual emoji polarity evidence	Aggregated polarity is contextual

2.1 Research Gap

Prior work commonly evaluates one method, one language, or one resource in isolation. Fewer systems preserve emoji explicitly while comparing sparse, subword, and contextual representations with identical label handling, artifact isolation, and user-facing diagnostics. The proposed study fills this engineering and evaluation gap for Marathi and Hindi.

III. PROPOSED WORK

The proposed system accepts a UTF-8 CSV file, validates the selected text and label columns, normalizes whitespace and Unicode, and preserves Devanagari as well as emoji sequences. Each emoji is retained in its original form and may be accompanied by a descriptive alias so that both glyph-level and lexical evidence can reach the models. The same stratified partition and label order are used across all model families, preventing optimistic comparisons caused by inconsistent preprocessing or test splits.

Emoji-aware processing and evaluation workflow



Figure 1 Proposed emoji-aware processing and evaluation workflow

3.1 Model Families

Model	Representation	Classifier	Primary strength	Main constraint
TF-IDF + SVM	Word and character n-grams	Linear SVM	Fast, compact, interpretable	Sparse and context-limited
fastText + LR	Subword-aware dense vectors	Logistic regression	Morphology and unseen forms	Weak word-order modelling
MuRIL	Contextual WordPiece embeddings	Transformer head	Indic context and transliteration	Slow and storage intensive

3.2 Training and Evaluation Design

Training produces separate artifacts beneath model-specific directories so that a new experiment cannot overwrite the original baseline. TF-IDF and fastText are suitable for CPU training. MuRIL uses supervised fine-tuning, validation-based model selection, early stopping, and device-aware inference; a compatible GPU reduces training time but is not required for inference. Accuracy measures overall correctness, while macro-F1 gives equal importance to all three

classes and is therefore central when class support differs.

- Input checks reject missing columns, blank text, unsupported labels, and malformed UTF-8 data.
- Prediction supports multiple sentences and reports the selected label with confidence values.
- Evaluation displays classification metrics, confusion matrices, model metadata, and version information.
- Model artifacts remain isolated from the original application and from one another.

3.3 System Architecture

The architecture separates the data layer, preprocessing, modelling, and Streamlit interface. This separation supports reproducibility and lets each model reuse the same normalized dataset while retaining its own vectorizer, classifier, metadata, and checkpoint. Cached loading prevents repeated transformer initialization during interactive use.

IV. IMPLEMENTATION AND EXPERIMENTAL SETUP

The implementation uses Python with scikit-learn for TF-IDF, SVM, and logistic regression; fastText-compatible subword vectors for dense features; Hugging Face tooling for MuRIL; and Streamlit for the interface. The dashboard exposes model selection, file upload, label mapping, training controls, batch prediction, confidence inspection, and metric views. The supplied demonstration corpus contains 120 balanced Marathi rows and is intended for functional validation rather than definitive generalization claims.

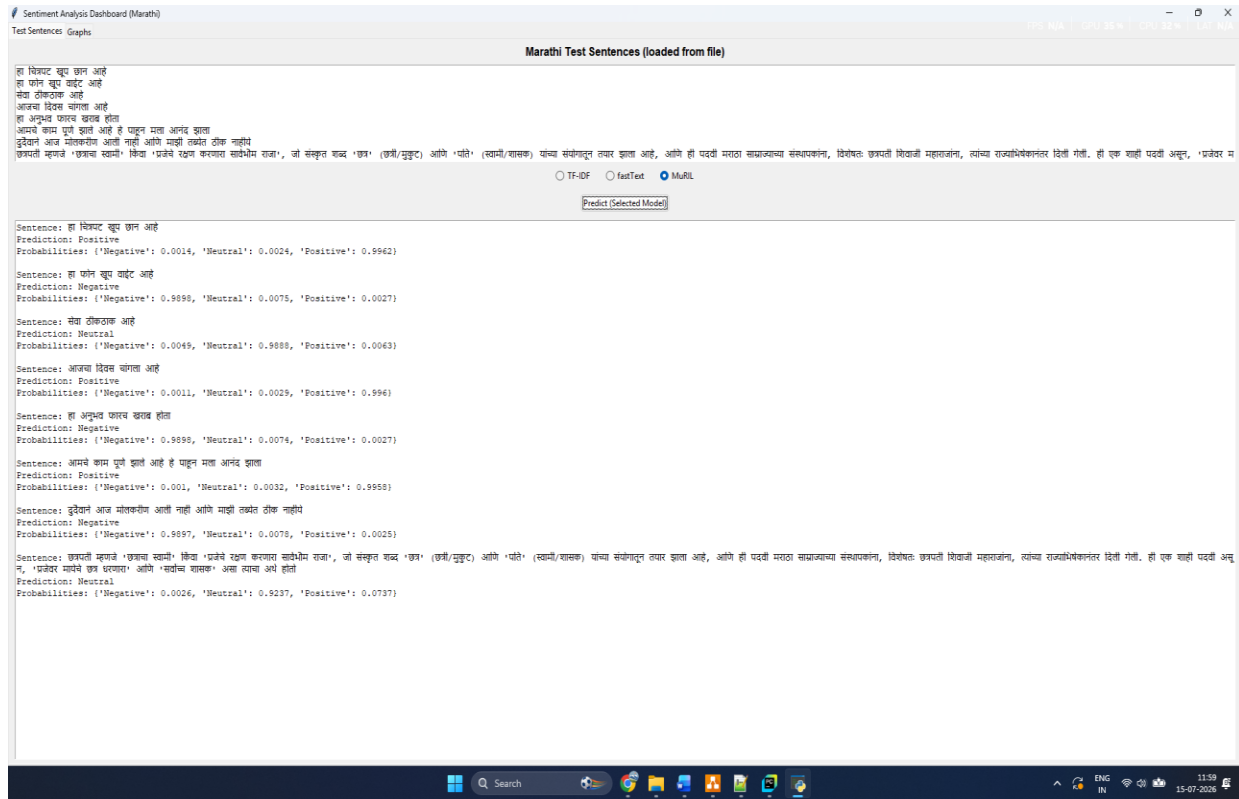


Figure 2 Batch prediction interface for Marathi sentiment analysis

Table 2 Functional validation summary

Test	Expected behavior	Outcome
Valid UTF-8 CSV upload	Rows and selected columns display correctly	Pass
Invalid label values	Training blocked with a clear message	Pass
TF-IDF training	Independent vectorizer, classifier, metadata	Pass
fastText training	Independent embedding and LR artifacts	Pass
MuRIL dependency check	Actionable error or disabled control	Pass
Batch classification	One prediction per non-empty sentence	Pass
Emoji visibility	Glyphs retained through input and output	Pass
Metric view	Accuracy, macro-F1, and confusion matrix shown	Pass

4.1 Evaluation Protocol

Reported benchmark values are taken from held-out evaluations in the supplied project record. The interface screenshots additionally show a perfect score on a small manually entered demonstration set. That demonstration confirms correct end-to-end behavior but is not treated as evidence of population-level accuracy. This distinction avoids conflating a sanity check with an independently sampled benchmark.

V. RESULTS AND DISCUSSION

5.1 Marathi Results

On the Marathi held-out evaluation, TF-IDF with SVM achieved 0.61 accuracy and 0.61 macro-F1. fastText improved both measures to 0.65, indicating that character-level subword information helps with Marathi morphology and spelling variation. MuRIL achieved the strongest result, with 0.804 accuracy and 0.803 macro-F1. The near equality of accuracy and macro-F1 suggests comparatively balanced performance across the three polarity classes in this test setting.

Table 3 Marathi held-out comparison

Model	Accuracy	Macro-F1	Interpretation
TF-IDF + Linear SVM	0.610	0.610	Reliable classical baseline
fastText + Logistic Regression	0.650	0.650	Subword features improve morphology handling
MuRIL	0.804	0.803	Best contextual and Indic-language representation

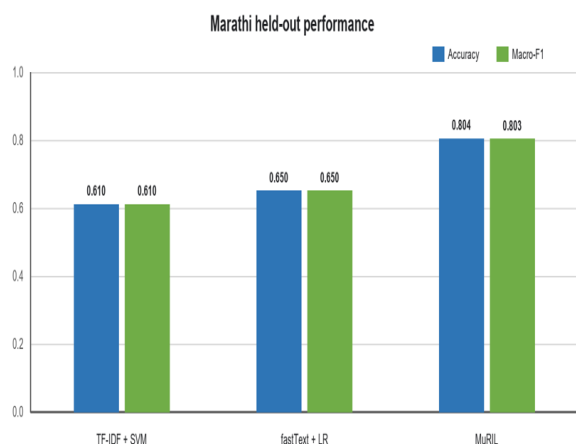


Figure 3 Accuracy and macro-F1 on the Marathi held-out set

5.2 Class-Level Interpretation

The supplied class-level summaries for the lighter baselines show that the neutral or middle class is generally the most difficult, with F1 near 0.56-0.59, while the strongest class reaches approximately 0.66-0.70. This pattern is consistent with ambiguous or mixed statements clustering around the neutral boundary. Emoji can sharpen some short messages, but a pictograph may also conflict with the surrounding text, especially in ironic usage.

MuRIL improves Marathi accuracy by 19.4 percentage points over TF-IDF and by 15.4 points over fastText. The gain supports contextual modelling, but the small corpus and possible domain overlap require cautious interpretation. For applications with strict latency or storage limits, the 0.65 fastText result may still be an attractive operating point.

5.3 Hindi Results and Cross-Language Discussion

The Hindi IITP evaluation is more difficult than the Marathi demonstration setting. TF-IDF achieved 0.3742 accuracy and 0.3681 macro-F1. fastText increased accuracy to 0.4387 but reduced macro-F1 to 0.3403, which indicates uneven class performance rather than a uniform improvement. MuRIL produced the best result at 0.6419 accuracy and 0.5507 macro-F1.

Table 4 Hindi IITP held-out comparison

Model	Accuracy	Macro-F1	Observed behavior
TF-IDF	0.3742	0.3681	Limited semantic and domain generalization
fastText	0.4387	0.3403	Higher accuracy but uneven class balance
MuRIL	0.6419	0.5507	Strongest contextual transfer

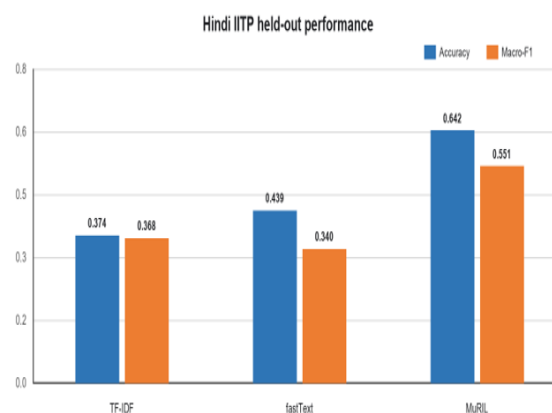


Figure 4 Accuracy and macro-F1 on the Hindi IITP held-out set

5.4 Comparative Discussion

Criterion	TF-IDF + SVM	fastText + LR	MuRIL
Training speed	Very high	High	Low on CPU
Inference speed	Very high	High	Moderate after caching
Context modelling	Low	Low to medium	High

Criterion	TF-IDF + SVM	fastText + LR	MuRIL
Unseen-word handling	Character n-grams	Subword vectors	WordPiece context
Interpretability	High	Medium	Low
Storage	Low	Medium	Very high

Across both languages, the ranking broadly favors MuRIL, but the size of the gain varies by dataset. The Hindi results also show why accuracy alone is insufficient: fastText predicts more instances correctly overall than TF-IDF while producing a lower macro-F1. Per-class review, confusion matrices, and domain-aware splits are therefore necessary before deployment. Statistical significance testing and repeated stratified runs should be added when larger datasets are available.

VI. CONCLUSION AND FUTURE WORK

6.1 Conclusion

This study demonstrates a complete emoji-aware sentiment-analysis workflow for Marathi and Hindi. It preserves Unicode symbols, enriches emoji as explicit evidence, applies consistent preprocessing, trains three independent models, and exposes predictions and metrics in a Streamlit dashboard. The experimental results establish a clear performance hierarchy: MuRIL is the strongest contextual model, fastText is the most capable lightweight semantic baseline, and TF-IDF with SVM remains the simplest reproducible benchmark.

For Marathi, MuRIL reached 0.804 accuracy and 0.803 macro-F1, compared with 0.65/0.65 for fastText and 0.61/0.61 for TF-IDF. For Hindi IITP, MuRIL reached 0.6419 accuracy and 0.5507 macro-F1. These results support Indian-language pre-training and contextual attention, while the larger Hindi error rates expose domain mismatch, class imbalance, and dataset difficulty. The interface-level demonstration confirms correct system operation but should not be interpreted as a substitute for held-out evaluation.

6.2 Limitations

- The 120-row demonstration corpus is too small to establish robust transformer generalization.

- Emoji aliases use coarse meanings and cannot fully represent irony, culture, conversation history, or emoji-text conflict.
- Domain and class distributions differ across Marathi and Hindi resources, limiting direct score comparison.
- Transformer training is slower, less interpretable, and more storage intensive than the two baselines.
- Single-split results do not quantify variance; repeated runs and statistical tests are still required.

6.3 Future Work

Future work should build a larger independently annotated corpus with inter-annotator agreement, language, domain, dialect, and code-mixing metadata. Difficult cases should include negation, sarcasm, mixed sentiment, transliteration, and explicit emoji-text disagreement. Repeated stratified cross-validation and confidence intervals would improve the reliability of model comparisons.

The modelling pipeline can be extended with domain-adaptive pre-training, class-balanced loss, calibrated probabilities, and parameter-efficient fine-tuning. Emoji representations should move from a fixed polarity lexicon to contextual role modelling that distinguishes indicator, intensifier, mitigator, and reverser behavior. Explainability methods such as token attribution can help users identify whether a prediction depends on words, emoji, or spurious features. Finally, model distillation and quantization can reduce MuRIL latency for local deployment, while privacy controls and drift monitoring are needed before using the system with real user content.

REFERENCES

- [1]. G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, pp. 513-523, 1988.
- [2]. B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, nos. 1-2, pp. 1-135, 2008.
- [3]. A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

- [4]. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019.
- [5]. P. Kralj Novak, J. Smailovic, B. Sluban, and I. Mozetic, "Sentiment of emojis," PLoS ONE, vol. 10, no. 12, 2015.
- [6]. Y. Zhou et al., "Cross-lingual analysis of literal and figurative emoji semantics," 2024.
- [7]. C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, pp. 273-297, 1995.
- [8]. P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," Transactions of the ACL, vol. 5, pp. 135-146, 2017.
- [9]. A. Conneau et al., "Unsupervised cross-lingual representation learning at scale," in Proc. ACL, 2020.
- [10]. D. Kakwani et al., "IndicNLP Suite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages," in Findings of EMNLP, 2020.
- [11]. S. Khanuja et al., "MuRIL: Multilingual representations for Indian languages," arXiv:2103.10730, 2021.
- [12]. A. Kulkarni, O. Mandhane, and R. Joshi, "L3Cube-MahaSent: A Marathi tweet-based sentiment analysis dataset," in Proc. WASSA, pp. 213-220, 2021.
- [13]. A. Pingle, A. Kulkarni, and R. Joshi, "L3Cube-MahaSent-MD: A multi-domain Marathi sentiment analysis dataset," in Proc. PACLIC, 2023.
- [14]. M. Akhtar et al., "A benchmark collection for sentiment analysis in Indian languages," in Proc. LREC, 2016.
- [15]. S. Kumar et al., "BHAAV: A benchmark dataset for Hindi sentiment analysis," in ACL Anthology Workshop Proceedings, 2019.
- [16]. F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.