

RomMa-FT: Static Word Embeddings for Romanized Marathi-English Code-Mixed Hate Speech Detection

Nitin Bhaskar Chavan and Badal Soni

Department of Computer Science and Engineering

National Institute of Technology Silchar, Assam, India

chavan21_rs@cse.nits.ac.in badal@cse.nits.ac.in

Abstract

In countries like India, people regularly use more than one language while expressing themselves on social media. Code-mixed Marathi-English text combines both languages in a single comment or post, with Marathi predominantly written in Roman script on social media. Yet all available Marathi NLP resources — such as MahaFT (Joshi, 2022) — are trained exclusively on Devanagari script, resulting in a complete out-of-vocabulary (OOV) problem on Romanized social media text. Although Chavan et al. (2023) addressed this gap with contextual transformer models (MeBERT, MeRoBERTa), static word embeddings for Romanized Marathi-English code-mixed text do not yet exist. We propose RomMa-FT, the first FastText-based static word embeddings trained on Romanized Marathi-English social media text using the L3Cube-MahaHate benchmark corpus of 37,500 hate-speech-annotated samples. Our embeddings leverage character n-gram subword features to handle non-standard spelling variants prevalent in Romanized Marathi (e.g., *khuup* / *khup* / *khoop*), achieving cosine similarities of 0.87 between canonical and variant forms — infeasible with Word2Vec, which returns zero vectors for unseen spellings. Extrinsic evaluation on binary hate speech detection with a BiLSTM classifier yields Macro-F1 of 0.8995, within 1.36% of fine-tuned MeBERT (0.9131) while using 128x fewer parameters (852K vs. 109M) and achieving 18.6x faster inference (1.41 s vs. 26.35 s). Nearest-neighbour analysis further reveals that RomMa-FT captures Marathi morphological inflections through character n-grams, grouping inflected forms of hate terms without any linguistic supervision.

1 Introduction

There are around 83 million people who use Marathi as their native language, making it the 3rd-largest spoken language in India. The increasing popularity of social media in multilingual societies

has resulted in code-switching — switching from one language to another within the same phrase. One such case is in Maharashtra, where speakers produce Romanized Marathi-English (RME): Marathi written in Roman letters alongside English. A typical example: “@BJP4Maharashtra neta madarachod chi shaharapramukh vhayachi layaki ahe ka.”

Despite Marathi being spoken by over 83 million people, NLP resources for Romanized Marathi-English remain limited. Current Marathi embeddings — MahaFT (Joshi, 2022), IndicFT (Kakwani et al., 2020), and Meta’s cc.mr.300 — were all created using Devanagari script, leaving them with 100% OOV on RME social media vocabulary. Words like *lavadya*, *chutiya*, *nahi*, and *khup* have no meaningful representation in any existing embedding model.

Chavan et al. (2023) made the first major contribution by releasing L3Cube-MeCorpus (10M sentences) and training MeBERT and MeRoBERTa — contextual BERT-based transformers for code-mixed Marathi-English. However, contextual transformers require substantial GPU resources for fine-tuning and offer limited interpretability of word-level representations. Static word embeddings remain essential for three reasons: (1) vocabulary-level semantic analysis — clustering, nearest-neighbour inspection, and interpretable hate lexicon construction; (2) lightweight deployment without GPU infrastructure; and (3) FastText’s character n-gram mechanism is uniquely suited to Romanized Marathi, where no spelling standard exists and variants of the same word are abundant.

We present RomMa-FT — the first static word embeddings trained on Romanized Marathi-English code-mixed social media text. Our contributions are:

- **RomMa-FT-SG and RomMa-FT-CB:** First FastText Skip-gram and CBOW embeddings for RME. On a curated reference vocabulary

of 38 Romanized Marathi-English words (Section 5.1), they reduce OOV from 100% (MahaFT) to 21.1%, with subword n-gram construction providing meaningful vectors even for the remaining OOV words.

- **Spelling variant robustness:** Character n-gram models achieve cosine similarity of 0.87 for *lavadyallavadyaa*, while Word2Vec returns zero vectors — directly addressing non-standard Romanized orthography (Section 5.2).
- **Comprehensive evaluation:** On the binary hate speech classification task of L3Cube-MahaHate (HOF vs. NOT), RomMa-FT-CB + BiLSTM achieves Macro-F1 of 0.8995, within 1.36% of MeBERT (0.9131) with 128x fewer parameters and 18.6x faster inference (Section 5.5).

2 Related Work

2.1 Marathi NLP Resources

Joshi (2022) presented L3Cube-MahaCorpus and MahaBERT — a 752M-token Marathi corpus with BERT models and MahaFT FastText embeddings, all trained on Devanagari script. Shanbhag et al. (2024) compared MahaFT against MahaBERT across multiple tasks, finding FastText remains competitive and computationally practical despite lower peak performance. These works establish Devanagari Marathi NLP but leave Romanized Marathi unaddressed.

2.2 Code-Mixed Marathi-English

Chavan et al. (2023) presented the first dedicated resources for code-mixed Marathi-English: L3Cube-MeCorpus (10M sentences, 5M Roman + 5M Devanagari) and transformer models MeBERT, MeBERT-Mixed, MeRoBERTa, and MeRoBERTa-Mixed, alongside benchmarks MeHate, MeSent, and MeLID (~12K tweets each). Despite this tremendous contribution, no static word embeddings were released — precisely the gap our work fills.

2.3 Code-Mixed Embeddings for Hindi-English

Nayak and Joshi (2022) released HingBERT and Hing-FastText for Hindi-English code-mixed text. Shruti and Soni (2024) showed Hing-FastText outperforms vanilla English FastText on code-mixed

Split	Total	HOF	NOT	Avg. Tok.
Train	30,000	15,000	15,000	18.3
Validation	3,750	1,875	1,875	18.1
Test	3,750	1,875	1,875	18.0
Total	37,500	18,750	18,750	18.2

Table 1: L3Cube-MahaHate Romanized dataset statistics. Label balance is 50/50 across all splits.

hate detection. Our work provides the Marathi-English equivalent — RomMa-FT is to Romanized Marathi-English what Hing-FastText is to Romanized Hindi-English.

2.4 Hate Speech Detection for Marathi

The L3Cube-MahaHate dataset (Patil et al., 2022) provides binary and fine-grained hate speech annotations for Marathi tweets. Subsequent work evaluated MahaBERT, MuRIL (Khanuja et al., 2021), and MeBERT on this benchmark. We use the Romanized version of MahaHate (37,500 samples) as both embedding training corpus and downstream evaluation benchmark, enabling direct comparability with prior results.

3 Dataset

We use the L3Cube-MahaHate Romanized dataset consisting of 37,500 Marathi-English code-mixed social media comments annotated for binary hate speech classification (HOF: Hate/Offensive; NOT: Non-Hate). The dataset is derived from L3Cube-MahaHate (Patil et al., 2022) by converting Devanagari text to Roman script using ITRANS transliteration while preserving naturally occurring English words, producing code-switched text that resembles real social media writing patterns. It is made available under CC BY-NC-SA 4.0 and is used here strictly for research purposes.

The resulting corpus is genuinely code-mixed: individual comments contain both Romanized Marathi words and English words within the same sentence, reflecting the natural Marathi-English code-switching behaviour observed on Indian social media. English tokens are retained in their original form (e.g., *government*, *seriously*, *party*), while the remainder are Romanized Marathi. Our nearest-neighbour analysis (Section 5.3) therefore reports results for both Romanized Marathi terms and English terms.

A salient characteristic of the dataset is orthographic variability: the same Marathi word appears

in multiple spellings across samples (e.g., *lavadya*, *lavadyaa*, *laavadya*, *lwdya*). It is important to clarify the origin of this variability. ITRANS transliteration is deterministic and produces a single canonical Roman spelling for any given Devanagari string; it does not itself introduce variants. The variability instead arises from two sources present in the original social media text. First, many comments in the source corpus were already written natively in Roman script by users, who romanize Marathi inconsistently as no standardized Roman orthography for Marathi exists. Second, users frequently employ expressive lengthening (*khuup* for *khup*), phonetic approximations (*khoop*), and deliberate obfuscation of slurs to evade automated moderation (*lwdya*). Since these variants exist as distinct surface forms in the corpus, they remain distinct after transliteration — motivating the use of character n-gram FastText embeddings, which assign similar vectors to orthographic variants of the same underlying word.

4 Methodology

4.1 Corpus Preparation

We construct an unsupervised training corpus by combining all three splits (37,500 texts), following standard embedding training practice — labels are not used. After removing URLs, @mentions, and purely numeric tokens, the corpus contains 686,286 tokens and 132,857 unique word types. Punctuation is stripped from tokens to prevent inflected forms with trailing marks (e.g., *lavadya.*, *lavadya*,) from being treated as distinct vocabulary entries.

4.2 RomMa-FT: FastText Embeddings

We train two FastText models (Bojanowski et al., 2017):

RomMa-FT-SG uses the Skip-gram method, which predicts surrounding context words given a target word. Skip-gram performs better on rare words — important for our corpus, which includes many Marathi slang terms that appear infrequently.

RomMa-FT-CB implements the CBOW algorithm, predicting a target word from its context. CBOW is computationally more efficient and works better with frequent words. Both models share hyperparameters: character n-grams (minn = 3, maxn = 6), dimension $d = 300$, 50 training epochs, and minimum word frequency of 2.

The key contribution of both models is the character n-gram mechanism. Each word w is repre-

sented as the sum of its character n-gram embeddings:

$$v(w) = \sum_{g \in G(w)} z_g \quad (1)$$

where $G(w)$ is the set of character n-grams of w and z_g is the n-gram embedding. This enables vector construction for unseen spelling variants by composing shared n-gram embeddings.

4.3 RomMa-W2V: Word2Vec Baseline

Word2Vec Skip-gram (Mikolov et al., 2013) is trained on the same corpus with identical hyperparameters ($d = 300$, window = 5, 50 epochs) as an ablation baseline. Word2Vec has no character n-gram features; OOV words receive a zero vector. Comparing RomMa-FT against RomMa-W2V isolates the contribution of subword information.

4.4 Downstream Classifier: BiLSTM

For extrinsic evaluation, we use a 2-layer Bidirectional LSTM (hidden size 128, dropout 0.3) followed by a linear classification head — a standard architecture in code-mixed hate speech detection. Input sequences are padded to 64 tokens. The same BiLSTM architecture is used for all embedding comparisons, ensuring performance differences are attributable solely to embedding quality.

We deliberately select a comparatively simple classifier rather than a more sophisticated architecture. The objective of this work is to isolate and quantify the contribution of the embeddings, not to maximise absolute task performance. A more expressive classifier could compensate for weaker embeddings, confounding the comparison between RomMa-FT, RomMa-W2V, and MahaFT. Holding the classifier fixed and minimal ensures that observed differences in Macro-F1 are directly attributable to embedding quality. We expect that pairing RomMa-FT with stronger architectures (e.g., attention-augmented BiLSTM or CNN-BiLSTM hybrids) would further improve absolute performance, and leave this to future work.

4.5 MeBERT Baseline

The `l3cube-pune/me-bert` model (Chavan et al., 2023) is fine-tuned on the same splits for 3 epochs ($lr = 2e-5$, batch size = 32), representing the contextual transformer upper bound. MeBERT is pretrained on 5M Roman-script Marathi-English sentences (70.9M tokens) — approximately 103x more data than our corpus.

Model	Script	OOV↓	Handling
MahaFT (Joshi, 2022)	Deva.	100.0%	Zero vec.
cc.mr.300 (Meta)	Deva.	~100.0%	Zero vec.
RomMa-W2V (ours)	Roman	21.1%	Zero vec.
RomMa-FT-CB (ours)	Roman	21.1%	N-gram
RomMa-FT-SG (ours)	Roman	21.1%	N-gram

Table 2: OOV rates on the 38-word Romanized Marathi-English reference vocabulary defined in Section 5.1. FastText constructs vectors for OOV words via character n-grams; W2V returns zero vectors.

5 Experiments and Results

5.1 OOV Rate Analysis

To quantify vocabulary coverage, we construct a reference vocabulary of 38 Romanized Marathi-English words drawn from high-frequency terms observed in our corpus. The reference set is deliberately balanced across four categories: 16 Marathi hate terms and slurs (e.g., *lavadya*, *chutiya*, *jhatu*), 13 common Marathi function and content words (e.g., *nahi*, *ahe*, *khup*, *bhai*), 5 English words that occur natively in the code-mixed text (e.g., *seriously*, *government*), and 4 social-media abbreviations (e.g., *wtf*, *lol*). The OOV rate reported in Table 2 is the percentage of these 38 words absent from each embedding model’s learned vocabulary. This reference set is intended as a diagnostic probe of whether a given model can represent the vocabulary that actually appears in Romanized Marathi-English social media, rather than as an exhaustive lexicon.

MahaFT achieves 100% OOV on this reference set, confirming complete failure on Romanized text: none of the 38 words exist in its Devanagari-trained vocabulary. All three RomMa models achieve 21.1% OOV (8 of 38 words). Crucially, for these remaining OOV words, FastText still constructs meaningful vectors via character n-grams (Equation 1), whereas Word2Vec produces zero vectors — a distinction quantified in Section 5.2.

5.2 Spelling Variant Robustness

Table 3 illustrates how FastText’s subword approach handles non-standard Romanized spellings. RomMa-FT-SG produces high cosine similarities between canonical forms and their variants by sharing character n-grams, while Word2Vec yields zero vectors (OOV) for all unknown variants. High similarities for *lavadya/lavadyaa* (0.87) and *madarachoda/madarachod* (0.89) confirm that RomMa-FT

Canonical	Variant	FT-SG↑	W2V
lavadya	lavadyaa	0.8666	OOV
lavadya	laavadya	0.7852	OOV
lavadya	lwdya	0.4328	OOV
chutiya	chutiya	0.8071	OOV
chutiya	chuutiya	0.7492	OOV
madarachoda	madarachod	0.8865	OOV
madarachoda	madarchod	0.3525	0.1829
madarachoda	madarachood	0.7841	OOV
khup	khuup	0.8137	OOV
nahi	naahi	0.4618	OOV

Table 3: Spelling variant cosine similarity. OOV = zero vector (Word2Vec has no subword mechanism).

effectively handles the orthographic variation characterised in Section 3.

5.3 Nearest Neighbour Analysis

Table 4 presents the top-5 nearest neighbours for probe words in RomMa-FT-SG, covering both Romanized Marathi and English terms. A key observation is that neighbours to Romanized Marathi words comprise predominantly their morphological inflections, sharing the same root. Neighbours to hate terms such as *chutiya* and *lavadya* include *chutiya* (genitive), *chutiya* (plural vocative), *lavadya* (dative), and *lavadya* (genitive feminine) — all morphologically inflected words carrying the same hate signal. This morphological awareness emerges via character n-grams without any supervision, and is extremely useful: it enables retrieval of semantic vectors for unseen inflected forms. For neutral Marathi words, *nahi* (not) groups with Marathi function words *pana* (but), *tara* (then), and *ahe* (is).

For English probe words, the behaviour differs in an informative way. The neighbours of *seriously* include *actually* — a semantically appropriate English discourse adverb — alongside orthographically related forms such as *series* and *sly*. This mixed pattern reflects the fact that English constitutes a minority of the corpus: English words receive fewer distributional co-occurrence signals than Marathi words, so character n-gram similarity contributes proportionally more to their representations. English terms embedded in RomMa-FT are therefore usable but weaker than in a dedicated English embedding space — an expected consequence of training on a Marathi-dominant corpus. The short abbreviation *bc* (2 characters) demonstrates poorer performance in both languages, illustrating a known FastText limitation for ultra-short tokens.

Probe	Cat.	Top-5 Nearest Neighbours
chutiya	Slur	hutiya, chutiyape, chutiyano, chutiyacha, chutiypa
lavadya	Slur	avadya, lavadyacha, lavadyata, lavadyala, lavadyachi
bc	Abbrev.	kbc, bcm, vergine, swag, admin
bhai	Neut. (mr)	bhaidooj, bhairava, bhaiye, bhaichya, bhaitada
nahi	Neut. (mr)	pana, tara, ahe, nahita, kahi
seriously	Eng. (en)	sly, vartulata, actually, series, z

Table 4: Nearest neighbours in RomMa-FT-SG for both Marathi (mr) and English (en) probes. Marathi neighbours are morphological inflections — an emergent property of character n-grams.

5.4 Cluster Cohesion

We compute average pairwise cosine similarity within semantic clusters to assess embedding quality. The hate word cluster (*lavadya*, *chutiya*, *gamda*, *harami*, *bc*, *mc*, *bullya*, *jhatu*) achieves a cohesion score of 0.2467, while the neutral word cluster (*bhai*, *yaar*, *chhan*, *khup*, *sangto*, *ahe*, *nahi*) achieves 0.3046. The higher neutral cohesion reflects the greater distributional consistency of Marathi function words compared to the diverse morphological forms of slang terms.

5.5 Hate Speech Classification Results

Table 5 presents the main extrinsic evaluation results on the binary hate speech classification task (HOF vs. NOT) defined in Section 3. RomMa-FT-CB + BiLSTM achieves Macro-F1 of 0.8995, outperforming RomMa-W2V + BiLSTM by 1.76 percentage points — directly quantifying the benefit of FastText’s subword mechanism. Fine-tuned MeBERT achieves 0.9131, a 1.36% margin over RomMa-FT-CB, while requiring 128x more parameters and 18.6x longer inference time.

5.6 Efficiency Analysis

The efficiency gap between RomMa-FT and MeBERT is substantial: RomMa-FT-CB uses 128x fewer parameters (852K vs. 109M), runs inference 18.6x faster (1.41 s vs. 26.35 s), and requires no GPU. This makes RomMa-FT suitable for resource-constrained deployments — mobile content moderation tools, browser plugins, and NGOs monitoring hate speech — where transformer fine-tuning is impractical.

6 Discussion

6.1 Complementarity with MeBERT

We present this work as explicitly complementary to, and not competitive with, MeBERT (Chavan et al., 2023).

We note that contextual transformers such as MeBERT can in fact yield word-level representations through several established techniques: aggregating subtoken embeddings for a word, running the model over single-word inputs and taking the final-layer representation, or averaging a word’s contextual embeddings across all its corpus occurrences. Our claim is therefore not that such analysis is impossible with MeBERT, but that static embeddings offer distinct practical advantages for it. Specifically: (1) vectors are precomputed and retrievable in constant time with no forward pass, enabling interactive vocabulary exploration over large lexicons; (2) no GPU is required at any stage; (3) character n-grams provide deterministic OOV handling for unseen spelling variants, whereas subword tokenisation splits variants inconsistently, so the aggregation strategy itself becomes a confounding design choice; and (4) a single canonical vector per word type avoids the ambiguity of choosing which contextual occurrence to treat as representative. Whether contextual word vectors extracted from MeBERT would outperform RomMa-FT on intrinsic vocabulary tasks is an open empirical question that we consider a valuable direction for future work.

The 1.36% F1 gap observed in Table 5 is thus a modest efficiency trade-off for a 128x parameter reduction, and the two families of representation serve different practical needs.

6.2 Morphological Inflection Discovery

An unexpected finding is RomMa-FT’s emergent morphological awareness. From the nearest-neighbour analysis, it is evident that even without morphological tagging, the model clusters inflected forms of Marathi words — *chutiyacha*, *chutiyano*, *chutiyape* — near the root form *chutiya*. This arises because of the overlap of character n-grams between the root and its inflected forms. This is very helpful for hate speech detection: an annotated hate

Model	Macro-F1 $\uparrow$	Accuracy	Parameters	Inference	GPU Req.
MahaFT + BiLSTM	~ 0.55	~ 0.55	852,162	~ 1.30 s	No
RomMa-W2V + BiLSTM	0.8819	0.8819	852,162	0.73 s	No
RomMa-FT-SG + BiLSTM	0.8981	0.8981	852,162	1.34 s	No
RomMa-FT-CB + BiLSTM	0.8995	0.8995	852,162	1.41 s	No
MeBERT $\dagger$ (Chavan et al., 2023)	0.9131	0.9131	109,483,778	26.35 s	Yes

Table 5: Classification results on the MahaHate Romanized test set (3,750 samples, binary HOF vs. NOT). $\dagger$ Contextual upper bound, fine-tuned on the same splits; requires GPU. All BiLSTM models use an identical architecture, so differences reflect embedding quality alone.

term automatically implies all its morphological inflections.

7 Conclusion

We proposed RomMa-FT, the first static FastText word embeddings for Romanized Marathi-English code-mixed social media text. RomMa-FT addresses the mismatch between Devanagari-only Marathi embeddings and Romanized social media text, reducing the OOV rate from 100% (MahaFT) to 21.1% on our reference vocabulary, with character n-grams providing meaningful vector construction for remaining OOV words. Spelling variant robustness analysis confirms cosine similarities of up to 0.89 between canonical and variant forms — practically very difficult to achieve with Word2Vec.

Extrinsic evaluation on MahaHate binary hate speech detection shows RomMa-FT-CB + BiLSTM achieves Macro-F1 of 0.8995 — 1.36% below fine-tuned MeBERT (0.9131) while using 128x fewer parameters and processing inputs 18.6x faster. Nearest-neighbour analysis reveals emergent morphological awareness without supervision. We position RomMa-FT as a lightweight, interpretable, GPU-free complement to transformer models for Romanized Marathi-English NLP. All trained models and the processed corpus will be released to the research community under CC BY-NC-SA 4.0.

Limitations

Overlap between embedding and evaluation data. Our embeddings are trained on the same corpus used for the downstream hate speech classification task. Although the embedding training stage is fully unsupervised and never observes labels — so no label leakage occurs and the comparison across embedding models remains internally valid — this setup does not reflect a realistic deployment scenario, in which embeddings would be trained on a large general corpus and applied to

unseen downstream data. We adopted this design because no sufficiently large independent corpus of Romanized Marathi-English text is publicly available. A stronger evaluation would train RomMa-FT on held-out Romanized Marathi-English text and evaluate on MahaHate without overlap; we regard constructing such a corpus as the highest priority for follow-up work.

Corpus size. Our corpus, comprising 37,500 sentences or 686K tokens, is far smaller than MeCorpus (10M sentences). Corpus size directly affects embedding quality, meaning RomMa-FT could perform substantially better with more Romanized Marathi-English data.

Short tokens. Short abbreviations such as *bc* or *mc* — frequently used as hate signals — are shorter than FastText’s minimum n-gram length ($\text{minn} = 3$), resulting in weaker representations. Incorporating abbreviation expansion as a preprocessing step is a promising remedy.

Single-task evaluation. Our extrinsic evaluation is restricted to binary hate speech classification. It remains an open question whether RomMa-FT transfers equally well to other Romanized Marathi-English tasks such as part-of-speech tagging, sentiment analysis, or named entity recognition. Morphology-sensitive tasks like POS tagging may particularly benefit from the emergent inflectional clustering reported in Section 5.3, whereas tasks requiring sentence-level contextual disambiguation may favour contextual models. Establishing a multi-task benchmark for Romanized Marathi-English is an important direction for future work.

Classifier simplicity. We use a deliberately simple BiLSTM classifier to isolate embedding quality. Absolute task performance would likely improve with stronger architectures, and the reported numbers should be read as a controlled comparison rather than as state-of-the-art results for the MahaHate task.

Ethics Statement

This work uses the publicly available L3Cube-MahaHate dataset (Patil et al., 2022), licensed under CC BY-NC-SA 4.0 and released for research purposes. The paper necessarily reproduces offensive Marathi terms in its analysis; these appear solely as objects of linguistic study and in no way reflect the views of the authors. Hate speech detection systems carry risks of misuse, including over-moderation of legitimate speech and disparate impact on particular communities. We emphasise that RomMa-FT is a representation resource rather than a deployable moderation system, and that any deployment should include human review and community-specific validation.

References

- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Tanmay Chavan, Om Gokhale, Aditya Kane, Sarang Patankar, and Raviraj Joshi. 2023. [My boli: Code-mixed Marathi-English corpora, pretrained language models and evaluation benchmarks](#). In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023*.
- Raviraj Joshi. 2022. L3Cube-MahaCorpus and MahaBERT: Marathi monolingual corpus, Marathi BERT language models, and resources. In *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*.
- Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLP Suite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961.
- Simran Khanuja, Diksha Bansal, Sujhant Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. [MuRIL: Multilingual representations for Indian languages](#).
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Efficient estimation of word representations in vector space](#). In *1st International Conference on Learning Representations, ICLR 2013*.
- Rajvardhan Nayak and Raviraj Joshi. 2022. L3Cube-HingCorpus and HingBERT: A code mixed Hindi-English dataset and BERT language models. In *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*.
- Hrushikesh Patil, Abhishek Velankar, and Raviraj Joshi. 2022. L3Cube-MahaHate: A tweet-based Marathi hate speech detection dataset and BERT models. In *Proceedings of the Third Workshop on Threat, Aggression and Cyberbullying (TRAC 2022)*, pages 1–9. Association for Computational Linguistics.
- Atharva Shanbhag, Maithili Shirke, and Raviraj Joshi. 2024. [Non-contextual BERT or FastText? a comparative analysis](#).
- S. Shruti and Badal Soni. 2024. [On importance of code-mixed embeddings for hate speech identification](#).
- Abhishek Velankar, Hrushikesh Patil, Amol Gore, Saloni Salunke, and Raviraj Joshi. 2022. [L3Cube-MahaHate: A tweet-based Marathi hate speech detection dataset and BERT models](#).