A Feature-Fusion Ensemble Approach for Tamil Hate Speech Detection

Sathasivam Nerujan and Kengatharaiyer Sarveswaran
University of Jaffna
sathasivamnerjan35@gmail.com and sarves@univ.jfn.ac.lk

Abstract

Detecting online toxicity in morphologically rich, low-resource languages like Tamil remains a major computational challenge. Standard transformer models often struggle with sub-word fragmentation, which can dilute the semantic intensity of regional insults and out-of-vocabulary slang. To mitigate this limitation, we train a multi-layer hybrid framework that fuses the deep contextual representations of L3Cube-TamilBERT with the character-level robustness of FastText embeddings. Our architecture leverages Last-4 Layers averaging and a dual pooling strategy (Mean + Max) to capture both global sentence intent and extract high-activation spikes of offensive cues typically lost in single layer representations. Experiments show that this hybrid model achieves a Macro-F1 of 0.7883, notably enhancing Hate Recall (0.7503) for detection of offensive content. Additionally, as reported by other studies, stacking ensemble achieves peak hate precision (0.9296), providing a high accuracy alternative for moderation scenarios requiring minimal false positives. By combining deep contextual hidden states with FastText embeddings, the proposed feature-fusion ensemble approach with multi-layer hybrid framework approach establishes a new benchmark for hate speech detection for Tamil.

Keywords: Tamil Hate Speech, Transformer Fusion, FastText, Dual Pooling

1. Introduction

Hate speech on social media causes direct harm to targeted communities and can amplify offline discrimination, making timely automated detection a critical societal and platform-level need. Surveys and empirical studies show that automatic hate speech detection is feasible, but remains error-prone and highly sensitive to domain shifts, annotation policies, and evolving language. Moreover, building reliable systems is a data intensive process that depends on large, carefully annotated corpora to capture nuanced, context dependent judgments and emerging forms of abusive language (Fortuna and Nunes, 2018).

Hate speech detection in Tamil remains challenging, apart from its low-resource nature, due to its agglutinative morphology, rich inflectional structure, and highly informal digital usage characterized by Tamil-English code-mixing, slang, and creative spellings (Maria Nancy et al., 2025). These properties create substantial vocabulary mismatch, where conventional transformer models often struggle (Xu, 2026) to capture culturally specific offensive expressions and morphologically complex word forms. In particular, sub-word tokenization can fragment region specific insults, weakening their semantic intensity and reducing detection sensitivity.

While pretrained transformer models provide strong contextual representations, relying solely on the final layer's `[CLS]` token is often insufficient for capturing the full semantic nuance of complex sentences. As noted by (Reimers and Gurevych, 2019), the default `[CLS]` output in BERT-based architectures frequently yields poor sentence-level embeddings, often underperforming compared to simple mean pooling. This limitation is particularly pronounced in Tamil hate speech detection, where localized offensive cues can be diluted within longer, agglutinative sequences or lost due to out-of-vocabulary slang. Consequently, recent work in toxic language moderation has moved toward hybrid feature-based approaches. For instance, (Iftikhar et al., 2025) demonstrates that fusing transformer ensembles with hybrid feature sets is essential for navigating the volatility of digital discourse. These theoretical and empirical gaps motivate our use of a more robust representation strategy, combining last-4 layers averaging with dual pooling and sub-word embeddings.

To address this, we propose a hybrid framework that integrates transformer-based contextual embeddings with FastText sub-word representations. The architecture incorporates last-4 layers averaging to enhance representational stability and a dual pooling strategy (mean and max) to capture both global sentence semantics and localized intensity peaks. Experimental results show that the proposed hybrid design achieves a Macro-F1 score of 0.7883, which is competitive with more complex ensemble-based approaches, while maintaining a moderate balance between hate recall (0.7503) and precision (0.6375). Unlike stacking-based models that strongly favor precision or large multilingual ensembles that prioritize recall at higher computational cost, our approach provides stable performance using a single

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 190–197
16 May 2026.

language-specific transformer, making it suitable for practical Tamil hate speech detection settings. Unlike prior work, our approach combines multi-layer transformer features, dual pooling, and FastText embeddings in a single framework, enabling better handling of semantic and morphological aspects of Tamil code-mixed text.

The primary contributions of this paper are: (1) We introduce a hybrid feature fusion architecture that combines transformer-based contextual embeddings with FastText sub-word representations to better handle Tamil's agglutinative morphology and informal, code-mixed social media text; (2) We propose a flexible multi layer representation strategy based on last-4 layers averaging and dual pooling (mean and max), and evaluate its performance alongside alternative layer averaging and pooling approaches; and (3) We present a systematic comparative analysis of single, hybrid, and ensemble modeling strategies, highlighting their precision, recall trade-off for Tamil hate speech detection.

2. Related Work

Recent research in Tamil hate speech detection has shifted from traditional machine learning algorithms, such as SVM and Random Forest, to transformer-based architectures to better handle code-mixed and morphologically rich social media text. Sangeetham et al. (Sangeetham et al., 2024) demonstrated this transition using the HASOC 2021 dataset (approximately 5.8K comments), where their transformer-based ensemble achieved an F1-score of 0.68, showing that conventional models struggle to capture the contextual nuances of Tamil digital discourse.

To address these limitations, multilingual transformer models such as mBERT and XLM-RoBERTa have been widely adopted. Karim et al. (Karim et al., 2025) leveraged ensembles of these models on the LT-EDI 2025 dataset (over 10K samples focused on caste and migration related hate speech), achieving a competitive F1-score of 0.803 through majority voting. In parallel, language-specific models such as L3Cube-TamilBERT have shown improved linguistic alignment; Roy et al. (Roy et al., 2025) reported an F1-score of 0.79 on a Tamil offensive language benchmark, outperforming multilingual baselines.

Despite these advances, handling out-of-vocabulary slang in Tamil-English code-mixed text remains challenging. Iftikhar et al. (Iftikhar et al., 2025) explored hybrid architectures on an 8K sample social media corpus, combining transformer contextual embeddings with FastText sub-word representations, achieving an F1-score of 0.77.

More recently, S et al. (S et al., 2025) proposed a multi-scale hybrid approach, fusing TF-IDF features at multiple scales with embeddings from five transformer models. Their three-level hierarchical ensemble integrates classical and contextual features across classifiers, achieving a peak F1-score of 0.818 on the LT-EDI 2025 shared task. The dataset used in this study is not publicly available, highlighting the challenges of benchmarking low-resource, code-mixed Tamil-English hate speech detection.

3. Approach

3.1. Dataset

We use the Tamil Offensive Speech Detection dataset, which consists of Tamil YouTube comments in native script, Romanized Tamil, and Tamil-English code-mixed text, annotated with binary labels (0 = non-offensive, 1 = offensive). The dataset is compiled from Tamil Wikipedia articles and shared-task resources, including the Dravidian-CodeMix HASOC @ FIRE 2020 shared task (Mandalam et al., 2020) and the Dravidian-CodeMix sentiment analysis task at FIRE 2020 (Chakravarthi et al., 2021). The full dataset is publicly available through Kaggle[1].

The original training split contains 6,649 offensive (label 1) and 21,226 non-offensive (label 0) instances. For evaluation, we utilize the official file `tamil_offensive_speech_val.csv` provided in the release (Hata, 2020), which comprises 1,684 offensive and 5,285 non-offensive instances. Table 1 summarizes the final distribution of these records after our preprocessing and duplicate removal steps. In addition to the Kaggle dataset, we also evaluate our model on the HASOC Code-Mixed Tamil dataset to ensure fair comparison with prior work, as several existing approaches are benchmarked on this dataset.

3.2. Preprocessing

We remove duplicate entries from both the training and validation splits, as the provided validation file also contains duplicates. Although it is common practice to remove emojis during preprocessing, we retain them because they convey affective cues that may be informative for offensive speech detection, as discussed in (Amalia et al., 2025).

3.3. Modeling Strategies

To systematically analyze the effectiveness of different representation learning paradigms for Tamil

[1] https://www.kaggle.com/datasets/eshikanahata/tamil-offensive-speech-detection

Table 1: Dataset label counts after preprocessing.

Split	Hate (1)	Non-hate (0)
Training	5953	18942
Validation	662	2105
Testing	1678	5181

hate speech detection, we experiment with multiple modeling strategies. These strategies are designed to evaluate the impact of fine-tuning, feature extraction, ensemble learning, and hybrid representation fusion.

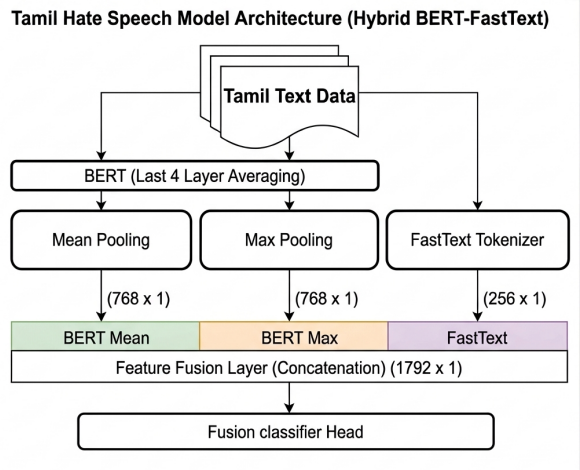

Figure 1: Overview of the proposed hybrid architecture.

3.3.1. Transformer Fine-Tuning

We fine-tune pretrained transformer models, including TamilBERT, MuRIL, and XLM-R, using a task-specific classification head. Each model is trained end-to-end with a softmax output layer for binary classification (Hate vs. Non-Hate), serving as strong neural baselines for comparison.

To combine the unique strengths of different architectures, we also developed a stacking-based ensemble. This is achieved by concatenating the sentence-level embeddings extracted from MuRIL and XLM-R. These fused representations are then processed by a Multi-Layer Perceptron (MLP) meta-classifier. By utilizing nonlinear hidden layers, the MLP learns complex interactions between the stacked features. This feature-level fusion approach yielded our peak Hate Precision of 0.9296, proving highly effective at balancing the specialized linguistic insights of both backbone models.

3.3.2. Hybrid Transformer FastText Models

To address the limitations of transformer tokenization on informal and out-of-vocabulary Tamil slang, we explore hybrid architectures that integrate transformer-based contextual embeddings with FastText sub-word representations. FastText leverages character-level n-grams to model subword patterns, enabling the model to capture both semantic context and morphological variations present in code-mixed Tamil text.

3.3.3. Proposed Feature-Level Fusion Architecture

Building on the hybrid framework, our approach incorporates last-4 layers averaging and dual pooling over transformer hidden states, followed by feature fusion with FastText embeddings. This fused representation is passed to an MLP classifier, enabling effective interaction between contextual and sub-word features for hate speech detection.

3.3.4. Transformer Backbone and Layer Averaging

We employ L3Cube-TamilBERT as the primary transformer encoder. To capture richer syntactic and semantic information, we extract hidden states from the last-4 transformer layers and compute their average.

Let H_i denote the hidden representation of the i^{th} transformer layer. The layer averaged representation is computed as:

$$H_{\text{avg}} = \frac{1}{4} \sum_{i=L-3}^{L} H_i$$

This multi-layer averaging strategy provides a more contextual representation, which is particularly beneficial for morphologically rich languages such as Tamil.

3.3.5. Dual Pooling Mechanism

Given the averaged hidden representation $H_{\text{avg}} \in \mathbb{R}^{T \times D}$, where T denotes the sequence length and D the embedding dimension, we apply a dual pooling strategy to obtain a fixed-length sentence representation.

The mean pooled representation is computed as:

$$\mu_{\text{pool}} = \frac{1}{T} \sum_{t=1}^{T} h_t$$

The max pooled representation is computed as:

$$M_{\text{pool}} = \max_{t=1}^{T}(h_t)$$

The final transformer-based sentence representation is obtained by concatenation:

$$V_{\text{BERT}} = [\mu_{\text{pool}}; M_{\text{pool}}]$$

Table 2: Performance comparison of evaluated models

Strategy Category	Model Architecture	Macro F1	Hate Recall	Hate Precision
Hybrid (Proposed)	TamilBERT + FastText (Layer Avg)	0.7883	0.7503	0.6375
Hybrid (Ensemble)	Hate-BERT + FastText (Layer Avg)	0.7888	0.7586	0.6346
Single Transformer	L3Cube-TamilBERT	0.7880	0.7384	0.6430
Single Transformer	mBERT	0.7639	0.6434	0.6432
Single Transformer	MuRIL	0.7770	0.6895	0.6456
Hybrid	MuRIL + FastText	0.7798	0.7682	0.6118
Meta-Classifier	MuRIL + XLM-R Stacking	0.7650	0.8008	0.9296
TF-IDF Fusion	MuRIL + XLM-R + TF-IDF	0.7004	0.5662	0.5373
Hybrid (Multilingual)	MuRIL + XLM-R + FastText	0.7867	0.8948	0.8960

This dual pooling strategy captures both the overall semantic context and localized high activation signals corresponding to offensive or toxic expressions.

3.3.6. FastText Sub-word Branch

The FastText branch generates sentence embeddings using character-level n-grams, capturing sub-word information for morphologically rich and out-of-vocabulary tokens. For each input sentence, a 300-dimensional embedding $V_{\text{FT}} \in \mathbb{R}^{300}$ is obtained.

To align with transformer-based features, V_{FT} is projected into a lower dimensional, task-specific space using a feed-forward layer with batch normalization, ReLU activation, and dropout:

$$H_{\text{FT}} = \text{Dropout}(\text{ReLU}(\text{BatchNorm}(WV_{\text{FT}} + b)))$$

where $W \in \mathbb{R}^{256\times300}$ and $b \in \mathbb{R}^{256}$. The resulting representation $H_{\text{FT}} \in \mathbb{R}^{256}$ is passed directly to the fusion stage.

3.3.7. Feature Fusion and Classification

The final fused feature representation is constructed by concatenating the dual pooled transformer outputs and the processed FastText embeddings:

$$V_{\text{fused}} = [V_{\text{BERT}} \parallel H_{\text{FT}}]$$

where $V_{\text{BERT}} \in \mathbb{R}^{1536}$ represent the mean and max pooled hidden states from the last four transformer layers, respectively, yielding a combined vector $V_{\text{fused}} \in \mathbb{R}^{1792}$.

This fused vector is passed to a Multi-Layer Perceptron (MLP) classifier with a hidden layer of 512 units, ReLU activation, and dropout. The final linear layer maps the 512-dimensional state to 2 output logits representing the classes (Hate vs. Non-Hate). During training, to directly combat dataset class imbalance, the model is optimized using a weighted cross-entropy loss function.

4. Results and Discussion

4.1. Performance Overview

Table 2 presents a comprehensive comparison of all evaluated architectures trained on the datasets detailed in Table 1. Given the class imbalance, the Macro F_1-score serves as our primary evaluation metric. Although the ensemble utilizing HateBERT[2] yielded the highest Macro F_1 of 0.7888, we did not select it as our best model. This backbone was pre-trained on YouTube and Twitter hate speech corpora, introducing a potential risk of data leakage or overlap with our test set. Therefore, our proposed `TamilBERT + FastText` framework (Macro F_1: 0.7883) stands as the most reliable architecture.

4.2. Comparative Analysis

4.2.1. Impact of Language-Specific Pretraining

Single L3Cube-TamilBERT achieved a Macro F1-score of 0.7880, outperforming the multilingual baseline mBERT (0.7639). This performance gap highlights the importance of language-specific pretraining, as Tamil only corpora allow the model to better capture agglutinative morphology, character level patterns, and syntactic constructions that are diluted in multilingual vocabularies.

4.2.2. Ablation Study and Architectural Analysis

To analyze the contribution of each architectural component, we evaluated multiple configurations using TamilBERT as a fixed backbone (Table 3). This systematic decomposition reveals how hybrid feature fusion and specific pooling strategies impact the model's ability to identify offensive content in complex linguistic environments.

The results indicate that the integration of FastText embeddings provides the most significant

[2] https://huggingface.co/mdosama39/tamil-bert-Caste-HateSpech_LTEDi-tamil

Table 3: Ablation study of architectural components using TamilBERT backbone

Model Strategy	Macro-F1	Prec.(Hate)	Rec.(Hate)
BERT + FastText	**0.7864**	0.6392	0.7390
BERT + Last-4 Layers Mean	0.7836	0.6621	0.6889
BERT + Dual Pooling	0.7755	0.5866	0.8153
BERT (Standard)	0.7647	0.5640	0.8302
BERT + Last Layer Mean	0.7356	0.5205	0.8254

overall performance gain. This effectiveness is largely attributed to its ability to mitigate out-of-vocabulary (OOV) slang and creative spellings in Tamil social media text. While standard transformer tokenizers often fragment such words into non-semantic sub-word units, FastText preserves essential morphological patterns through character n-grams.

Furthermore, leveraging the Last-4 Layers Mean enhances representation stability. By moving beyond the final hidden state which can be overly specialized to the pretraining objective the model captures deeper contextual signals. While dual pooling improves recall by emphasizing localized high-activation features, it introduces a minor reduction in precision due to increased sensitivity to noisy signals. Overall, the fusion of FastText with multi-layer contextual embeddings creates the most balanced representation for this morphologically rich and code-mixed language.

4.2.3. Dual Pooling and Multi-Layer Representation

Table 4: Effect of different pooling strategies using TamilBERT backbone

Pooling Method	Macro-F1
CLS Token	0.7647
Mean Pooling	0.7356
Dual Pooling (Mean + Max)	0.7755

Table 5: Effect of layer combination strategies using TamilBERT backbone

Layer Strategy	Macro-F1
Last Layer	0.7356
Last-4 Layers	0.7836

To isolate the contribution of each component, we evaluated pooling strategies and layer combination methods independently using TamilBERT as a fixed backbone. As shown in Table 4, dual pooling outperforms both CLS token and mean pooling, demonstrating its ability to capture both global sentence semantics and localized high activation features. This suggests that relying solely on the `[CLS]` representation is insufficient for modeling sentence-level meaning, consistent with prior findings (Reimers and Gurevych, 2019).

Similarly, Table 5 shows that combining the last-4 transformer layers significantly improves performance over using only the final layer, highlighting the importance of leveraging richer contextual representations. We do not extend the aggregation beyond the last-4 layers, as prior work (Tenney et al., 2019) has shown that higher transformer layers capture more task-relevant semantic information, while incorporating lower layers provides diminishing returns due to redundancy in lower-level features. This observation is consistent with existing findings and supports our design choice of restricting layer fusion to the last-4 layers.

Although dual pooling and multi-layer fusion provide complementary benefits, their improvements are not strictly additive indicating partial redundancy between learned features, suggesting partial feature overlap. Overall, these results emphasize that careful selection of pooling and layer strategies is critical for robust representation learning.

4.2.4. Qualitative Analysis

Table 6: Qualitative examples where the hybrid model correctly predicts non-hate cases

Text	Label
dhanush looks like gramathu vayasana thathaa	0
தமிழனை தமிழனை வைத்து அழிங்கிர சூழ்ச்சிதான் இது.....சாதி.....பா.ரஞ்சித் எதிர் மோகன்.....ஒரு நாள் எல்லா தமிழ் சமுகமும் கண்ணிர் விடும்....தன்னை தானே அழித்துகொண்டோம் என்று....	0
vvedalam teaser likes 145k but bairavaa teaser likes 171k beated vedalam teaser in just 2 days!!!!!	0

As shown in Table 6, the proposed hybrid model correctly classifies challenging non-hate examples that are misclassified by other architectures evaluated in Table 3. These errors typically arise due to informal language, cultural references, or noisy social media patterns.

Transformer only models tend to over predict offensive content due to fragmented tokenization, while FastText only or partial fusion models fail to capture deeper contextual cues. In contrast, the hybrid architecture effectively combines sub-word information with multi-layer contextual representations, allowing it to handle such nuances more accurately.

These examples further validate that the full hybrid configuration is more robust than its individual components, as reflected in the ablation results.

Table 7: Comparison on HASOC Code-Mixed Tamil dataset (SubTask 2)

Model / Study	Methodology	Macro F1
Our Hybrid Model	TamilBERT + FastText + L4 Fusion	0.8955
Siva@HASOC	XLM-R + mBERT (Ensemble + Transliteration)	0.8800
MUCIC@HASOC	Logistic Regression / BERT	0.6790
PSG@HASOC	MuRIL + Mean Pooling	0.6700

4.2.5. Sensitivity to Code-Mixed vs Pure Tamil Inputs

Table 8: Performance comparison across Pure Tamil and Code-Mixed inputs

Input	Macro-F1	Prec.(Hate)	Rec.(Hate)	Accuracy
Pure Tamil	0.7790	0.5301	0.7719	0.8813
Code-Mixed	0.7809	0.6172	0.7730	0.8216

Table 8 shows the model's performance across Pure Tamil and Code-Mixed inputs. Despite differences in data distribution, the model achieves nearly identical Macro-F1 scores, indicating low sensitivity to input type. While precision is higher for code-mixed text, recall remains consistent across both settings. Overall, this demonstrates that the model generalizes effectively to both pure Tamil and code-mixed social media content.

4.2.6. Fair Comparison with Existing Models

To ensure a fair comparison with existing models on Tamil hate speech detection, we first evaluated a publicly available pretrained model, Hate-speech-CNERG/deoffxlmr-mono-tamil (Saha et al., 2021). Although this model reports a Weighted F1-score of 0.76 (and 0.78 with ensembling), we observed an high Macro F1-score of 0.98 when evaluated on our test set, indicating that the model was likely trained on overlapping data.

To avoid such data leakage and ensure a consistent evaluation setting, we instead adopt the HASOC Code-Mixed Tamil dataset (SubTask 2), which is widely used in prior work. We then apply our hybrid methodology on the same dataset and compare it with previously reported results.

As shown in Table 7, our hybrid model achieves the highest Macro-F1 score (0.8955) with TamilBERT, outperforming prior approaches on the same dataset. While Siva@HASOC (Sai and Sharma, 2020) reports competitive results using multilingual ensembles with transliteration, details of their Exact test split are not publicly available, limiting reproducibility.

Compared to earlier systems such as MUCIC (Balouchzahi et al., 2022) and PSG@HASOC ((Benhur and Sivanraju, 2021)), our approach shows clear improvements by integrating subword representations and multi-layer contextual features, enabling more robust handling of code-mixed and morphologically complex Tamil text. Unlike prior work, our method does not rely on external transliteration or ensemble complexity, making it more suitable for real world deployment.

Overall, these results demonstrate that feature-level hybrid fusion provides a more effective and efficient alternative to both traditional models and complex ensemble strategies.

4.2.7. Class-Level Trade-offs

Our models exhibited clear precision recall trade-offs. The proposed hybrid achieved strong Hate Recall (0.7503), enabling effective detection of offensive content, while the MuRIL + XLM-R stacking model attained the peak Hate Precision (0.9296), making it suitable for moderation scenarios requiring minimal false positives.

4.2.8. Why TF-IDF Fusion Underperformed

The MuRIL + XLM-R + TF-IDF approach performed poorly (Macro-F1: 0.7004) due to the incompatibility between sparse TF-IDF features and dense transformer embeddings, which introduces lexical noise in morphologically rich Tamil text.

4.2.9. Summary

Overall, the results confirm that combining multi-layer transformer representations, dual pooling, and FastText embeddings outperforms both single-model baselines and traditional decision-level ensembles for Tamil hate speech detection.

5. Conclusion

This work evaluates transformer-based, hybrid, and ensemble architectures for Tamil hate speech detection, using Macro-F1 as the primary metric. The proposed L3Cube-TamilBERT + FastText hybrid framework, which combines deep contextual embeddings with FastText representations achieves F1 score of 0.7883 While stacking ensembles and multilingual hybrids in Table 2 achieve better hate precision and hate recall, they compromise overall stability showing low F1

scores. These results demonstrate that the proposed feature-fusion ensemble approach which integrate multi-layer representations with dual pooling offer a more reliable and scalable approach for low-resource languages like Tamil, particularly when benchmark datasets are limited or not publicly available.

6. Bibliographical References

References

Junita Amalia, Sarah Rosdiana Tambunan, Susi Eva Maria Purba, and Walker Valentinus Simanjuntak. 2025. Enhancing hate speech detection: Leveraging emoji preprocessing with bi-lstm model. *Journal of Information Systems and Informatics*, 7(2). Published June 2025.

Fazlourrahman Balouchzahi, Sepideh Bashang, Grigori Sidorov, and H. Shashirekha. 2022. Comata oli-code-mixed malayalam and tamil offensive language identification.

Sean Benhur and Kanchana Sivanraju. 2021. Psg@hasoc-dravidian codemixfire2021: Pretrained transformers for offensive language identification in tanglish. *CoRR*, abs/2110.02852.

Paula Fortuna and Sergio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys*.

A. Iftikhar et al. 2025. Beyond words: A hybrid transformer-ensemble approach for detecting hate speech and offensive language on social media. *PeerJ Computer Science*.

S. F. Karim et al. 2025. Cuet_blitz_aces@lt-edi-2025: Leveraging transformer ensembles and majority voting for hate speech detection. In *Proceedings of the 5th Workshop on Speech, Vision, and Language Technologies for Dravidian Languages (ACL Anthology)*.

Thomas Mandl, Sandip Modha, Anand Kumar M, and Bharathi Raja Chakravarthi. 2021. Overview of the hasoc track at fire 2020: Hate speech and offensive language identification in tamil, malayalam, hindi, english and german. In *Proceedings of the 12th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '20, page 29–32, New York, NY, USA. Association for Computing Machinery.

C Maria Nancy, N Radha, and R Swathika. 2025. SSN_IT_HATE@LT-EDI-2025: Caste and migration hate speech detection. In *Proceedings of the 5th Conference on Language, Data and Knowledge: Fifth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 84–89, Naples, Italy. Unior Press.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

B. Roy et al. 2025. Lexilogic@dravidianlangtech 2025: Multimodal hate speech detection in dravidian languages. In *Proceedings of the DravidianLangTech@NAACL 2025 (ACL Anthology)*.

Ganesh Sundhar S, Durai Singh K, Gnanasabesan G, Hari Krishnan N, and Mc Dhanush. 2025. Wise@LT-EDI-2025: Combining classical and neural representations with multi-scale ensemble learning for code-mixed hate speech detection. In *Proceedings of the 5th Conference on Language, Data and Knowledge: Fifth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 54–62, Naples, Italy. Unior Press.

Debjoy Saha, Naman Paharia, Debajit Chakraborty, Punyajoy Saha, and Animesh Mukherjee. 2021. Hate-alert@DravidianLangTech-EACL2021: Ensembling strategies for transformer-based offensive language detection. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 270–276, Kyiv. Association for Computational Linguistics.

Siva Sai and Yashvardhan Sharma. 2020. Siva@ hasoc-dravidian-codemix-fire-2020: Multilingual offensive speech detection in code-mixed and romanized text. In *FIRE (Working Notes)*, pages 336–343.

S. Sangeetham et al. 2024. Enhanced detection of hate speech in dravidian languages in social media using ensemble transformers. *Interdisciplinary Journal of Information, Knowledge, and Management*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. *CoRR*, abs/1905.05950.

Ning Xu. 2026. Tokenization and morphological fidelity in Uralic NLP. *arXiv preprint*. ArXiv:2602.04241.

(Mandl et al., 2021)

7. Language Resource References

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Shardul Suryawanshi, Navya Jose, Elizabeth Sherly, and John P. McCrae. 2021. Overview of the track on sentiment analysis for dravidian languages in code-mixed text. In *Proceedings of the 12th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '20, page 21–24, New York, NY, USA. Association for Computing Machinery.

Hata, Eshika Nahata. 2020. *Tamil Offensive Speech Detection*. Kaggle. Kaggle dataset; accessed 2026-02-27.

Asrita Venkata Mandalam, Yashvardhan Sharma, Bharathi Raja Chakravarthi, et al. 2020. Overview of the hasoc-dravidiancodemix shared task on offensive language identification in code-mixed text. In *Proceedings of the 12th Forum for Information Retrieval Evaluation (FIRE 2020)*, pages 55–59.