FLAICOL: Flip-Point-Led Augmentation for Imbalanced Code-Mixed Offensive Language Detection

Danish Mohammed and Vidhya Kamakshi
National Institute of Technology Calicut, Kozhikode - 673601, Kerala, India.
{danish_m240681cs,vidhyakamakshi}@nitc.ac.in

Abstract

Hate speech detection in low-resource, code-mixed languages is a challenging task as people often switch between scripts and languages in a single post. Code-Mixed scripts can take the form of explicit slurs, subtle insults, or fragmented abuse, and is often hidden by spelling variants and Romanized script. These datasets are also subjected to class imbalance with hate speech being a minority class of interest. To mitigate the imbalance, targeted data augmentation of minority class samples can help learn better representations to aid hate speech detection despite the naturally expected imbalance. We propose FLAICOL, a flip-point method which identifies the minimal embedding perturbation that moves an input across the decision boundary, map it back to discrete text, and retrain on those focused examples. Empirical results show that these interpretable augmentations strengthen Transformer classifiers on low-resource, code-mixed low resource hate datasets (Experiments were conducted on the Tamil-English, Malayalam-English, and Kannada-English splits in the Dravidian CodeMix Benchmark).

1 Introduction

Hate speech on social media has grown sharply in recent years, with platforms reporting a steady rise in abusive and offensive content. In multilingual regions such as India, users often mix native scripts (e.g., Tamil, Malayalam) with Latin tokens (Chakravarthi et al., 2022) in the same message. This mixing leads to lexical variation alternate spellings like "school" vs. "skool" and script alternation that monolingual detectors fail to handle. Simple keyword filters and classical machine-learning models struggle when words appear in unexpected forms or scripts. Early works applied SVMs, Naïve Bayes, and decision-tree classifiers (Das et al., 2023) to code-mixed text but yielded low accuracy because these models cannot learn subword patterns or contextual cues across scripts. Character-level CNNs (Rani et al., 2020) later proved effective at modeling noisy spellings, boosting F1 score on mixed datasets. Transliteration schemes (Roy and Kumar, 2025) that map all text to a single script reduced vocabulary size and improved recall on code-mixed data. Adversarial perturbations in transformer embeddings (Liu et al., 2020) further increased robustness to orthographic noise, showing gains in F1 scores on benchmark splits. Lexicon-injected transformer methods (Sariyanto et al., 2025) better handle rare offensive terms, yielding recall improvements. Cohen et al. (Cohen et al., 2023) leverage back-translation and LLM capabilities for data augmentation to enhance hate speech detection accuracy. This is a general augmentation strategy, unlike our proposed framework, FLAICOL, which performs boundary-aware augmentation.

While these approaches improve robustness, they often lack interpretability (Sharma et al., 2021; Kamakshi and Krishnan, 2023). To improve both performance and transparency, we propose FLAICOL, a framework leveraging a flip-point (a neighboring instance that is close to an instance of interest but flips the prediction) method (Yousefzadeh and O'Leary, 2020) that adapts the closest-flip-point and homotopy framework for pretrained Transformer classifiers. We compute a homotopy set and trace the homotopy back to the original classifier while optimizing small token-embedding perturbations and converting validated continuous solutions into discrete, targeted augmentation examples. To mitigate data scarcity and class imbalance (Thandil et al., 2025), these flip-derived examples are used to enrich offensive samples before fine-tuning. We evaluate our approach on Tamil–English, Malayalam–English, and Kannada-English corpora from the DravidianCodeMix benchmark (Chakravarthi et al., 2022), focusing on improving performance

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 21–31

for low-resource, code-mixed hate-speech detection. Our results and analysis aim to support the development of more reliable and fair moderation tools and to help create safer online spaces for multilingual communities.

2 Related Work

Early efforts (Das et al., 2023) in hate speech detection relied heavily on traditional machine learning classifiers such as SVMs, Naïve Bayes, and decision trees. These methods struggled to capture contextual nuances and cross-lingual dependencies inherent in code-mixed text. Rani et al. (Rani et al., 2020) demonstrated that character-level CNNs outperformed these classical models by modeling subword patterns, achieving significantly higher F1 scores. Similarly, Saumya et al. (Saumya et al., 2021) showed that in very low-resource settings, character-level n-gram TF–IDF features combined with Naïve Bayes or logistic regression outperformed deep models like BERT (Devlin et al., 2019) and ULM-FiT (Joseph and Joshi, 2024). Roy & Kumar (Roy and Kumar, 2025) further reported that transliterating Telugu–English code-mixed text into a unified script before modeling improved accuracy to 75%, reducing vocabulary variation.

The introduction of adversarial techniques marked an important advance in strengthening multilingual Transformer models. Liu et al. (Liu et al., 2020) fine-tuned ERNIE 2.0 on Hindi–English data and injected adversarial perturbations into XLM-R embeddings, producing F1 gains of up to 5% across ensemble folds. Their results show that small, targeted perturbations in embedding space can act as an effective regularizer. Like adversarial techniques, lexicon-driven methods improve interpretability by tying model decisions to curated word lists and linguistically informed features.

Lexicon-driven methods brought interpretability into hate speech detection. Sariyanto et al. (Sariyanto et al., 2025) introduced the VAD-Baseline method, which maps words to valence, arousal, and dominance scores, using gated summation to identify hateful content. Pamungkas & Patti. (Pamungkas and Patti, 2019) integrated LSTM-based sequence models with HurtLex features, improving recall across languages but experiencing significant performance drops in cross-domain testing. Recent hybrid models (Al Nahian et al., 2025) injected lexicon-derived vectors into transformer architectures like mBERT (Devlin et al., 2019), MuRIL (Khanuja et al., 2021), and XLM-R (Conneau et al., 2020), using concatenation or Bi-LSTM layers, achieving good macro-F1 scores on Hindi–English and on Tamil and Malayalam.

The rise of large pretrained transformers significantly improved code-mixed hate speech detection. Varma et al. (Varma et al., 2022) enhanced Malayalam–English detection by pairing BERT embeddings with a logistic regression head. Chakravarthi et al. (Chakravarthi et al., 2022) demonstrated that XLM-RoBERTa consistently outperforms other multilingual encoders on the DravidianCodeMix dataset. Kumar et al. (Kumar et al., 2020) investigated multilingual joint fine-tuning of mBERT (Devlin et al., 2019), mDistilBERT (Sanh, 2019), and XLM-R (Conneau et al., 2020) for hate speech, sentiment, and event detection, finding moderate gains in noisy conditions.

Cross-lingual transfer and ensembling have further boosted performance across several studies. Ghosh & Senapati (Ghosh and Senapati, 2022) evaluated monolingual and multilingual models on Indic languages, reporting MuRIL's (Khanuja et al., 2021) weighted F1 scores above 0.90 for Bengali and around 0.83 for Hindi. Kakati & Dandotiya (Kakati and Dandotiya, 2024) combined a DConv-BLSTM with MuRIL (Khanuja et al., 2021) for Hinglish (Hindi-English), Tamil–English, and Malayalam-English, achieving F1 scores above 0.94. A few of the recent works (Goswami et al., 2023; Raihan et al., 2023, 2024) consider a realistic code-mixing scenario in which more than two languages are mixed in a single instance, and the authors assess the performance of various transformer architectures on such data.

The availability of targeted datasets has also enabled more focused studies: Anbukkarasi & Varadhaganapathy (Anbukkarasi and Varadhaganapathy, 2023) took a Tamil–English corpus and reported an F1 score of 0.81 in Bi-LSTM, while Kumar et al. (Kumar et al., 2025) introduced LexiLogic, a multimodal system that uses fine-tuned BERT variants with back-translation and synonym replacement to obtain high macro-F1 scores. Kavatagi et al. (Kavatagi and Rachh, 2025) developed HASTIKA, a fine-grained Kannada-English hate and target identification system that achieved an F1 score of around 0.78.

Despite these advances, a major bottleneck that remains is the scarcity of large, diverse annotated corpora for code-mixed hate speech, especially for low-resource languages. The Dravidi-

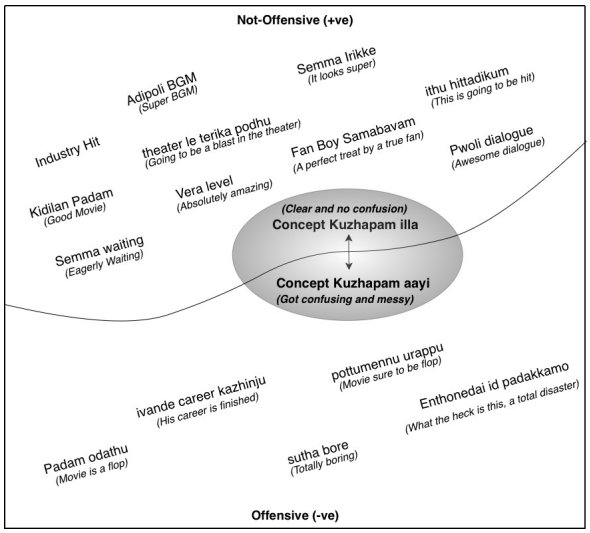

Figure 1: Conceptual embedding-space illustration of FLAICOL, the proposed homotopy flip-point idea. The decision boundary separates Not-Offensive (positive) and Offensive (negative) regions, the shaded oval indicates the homotopy search region and the arrow denotes the minimal semantics-preserving perturbation $\delta_{\min}$ that produces a flip point. The translation of the code-mixed instances is provided in brackets to facilitate global researchers.

anCodeMix dataset introduced by Chakravarthi et al. (Chakravarthi et al., 2022) provides manually labeled code-mixed examples and is a valuable training resource, but its severe class imbalance (Thandil et al., 2025) continues to limit model robustness and generalization. Our proposed methodology, FLAICOL, aims to address the imbalance by augmenting the minority (offensive) class using flip points.

3 Proposed Methodology

The proposed approach FLAICOL is inspired by the flip-point and homotopy framework (Yousefzadeh and O'Leary, 2020), adapted for pretrained Transformer classifiers and discrete text by using a bias homotopy on the classifier head and searching in token embedding space.

As illustrated in Figure 1, our approach searches for minimal semantic perturbations that move an example across the classifier decision boundary. These flip points are decoded into discrete token edits and used for data augmentation.

3.1 FLAICOL, a Flip-point framework

A *flip point* is a minimally perturbed input that lies on the decision boundary between two classes. Formally, for an input x predicted as class i and a chosen target class j, the *closest flip point* $x^\star$ is defined as the solution to the constrained optimization

$$\begin{aligned} x^\star = \arg\min_{x'} \|x' - x\|_2^2 \\ \text{s.t.} \quad z_i(x') = z_j(x') \\ z_i(x') \geq z_k(x') \quad \forall k \notin \{i, j\} \\ x' \in [L, U] \end{aligned} \tag{1}$$

where $z(\cdot)$ denotes the model logits (pre-softmax) and $[L, U]$ denotes feasible input bounds.

Because textual inputs are discrete, we adapt the homotopy idea for Transformers used for binary classification by first computing a minimal change to the classifier bias that makes the given example feasible. Given an encoder representation $h(x)$ and the classifier head parameters (W, b), the homotopy set is obtained by solving

$$\begin{aligned} b_{\text{homo}} = \arg\min_{b'} \|b' - b_{\text{orig}}\|_2^2 \\ \text{s.t.} \quad (h(x)W^\top + b')_i \leq (h(x)W^\top + b')_j \end{aligned} \tag{2}$$

Once a feasible bias b_{homo} is found, we trace (interpolate) the bias back to the original bias b_{orig} and re-solve for a nearby flip at each interpolation step.

To make the constrained search tractable for tokenized text we perform optimization in continuous token-embedding space. Let E_0 denote the original sequence of token embeddings for x, and let Δ denote small continuous offsets applied to a selected subset of token positions. As a practical surrogate for the constrained problem (equation (1)), we optimize the following penalty objective:

$$\mathcal{L}(\Delta) = \|\Delta\|_2^2 + \alpha(z_i(E_0 + \Delta) - z_j(E_0 + \Delta))^2 \tag{3}$$

where $\alpha > 0$ is a penalty weight controlling the enforcement of minimal perturbation to the instance under consideration and flipping of the classification decision. After optimizing (3) , the perturbed embedding $E_0 + \Delta$ is treated as a flip. The continuous solution is then mapped back to discrete tokens by nearest-neighbor lookup in the model's embedding matrix. The decision obtained on the mapped augmented instance is then validated on the original classifier (to ensure that minority-class representative examples are increased through the flip-point-led augmentation).

4 Experimental Setup

Experiments were conducted using Google Colab with GPU acceleration (NVIDIA T4, 16GB VRAM). All models were implemented in PyTorch using the HuggingFace transformers framework. Mixed-precision (FP16) training was enabled when CUDA was available.

We evaluate two pretrained multilingual Transformer backbones, namely XLM-RoBERTa-base (Kodali et al., 2025) and MuRIL-base (Khanuja et al., 2021).

Both models were fine-tuned for binary sequence classification (Not-Offensive vs Offensive).

4.1 Flip-Point Configuration

The following configuration was adopted for both backbones:

- Maximum Sequence Length = 64
- α = {1, 10, 100}
- Learning rate = 10^{-2}
- Number of Homotopy Iterations = 6
- Batch Size = 64
- Optimizer: Adam (Kingma and Ba, 2015)
- Random seed: {1, 2, 3, 42 }.

The implementation can be accessed from `https://github.com/FLAICOL/FLAICOL.git`

4.2 Dataset

We utilize the DravidianCodeMix corpora, which are specifically curated for hate speech detection in code-mixed text. The datasets are prepared with two label categories: "Not-Offensive" and "Offensive". The data consists of user-generated comments from YouTube, reflecting real-world scenarios where Dravidian languages are mixed with English (code-mixing). The details of the specific splits that we use are given below:

- *Tamil–English:* This corpus consists of 42,133 examples. It shows a notable class imbalance, with 31,808 "Not-Offensive" instances and 10,325 "Offensive" instances.
- *Malayalam–English:* This corpus consists of 18,403 examples. It exhibits a severe class imbalance, with 17,697 "Not-Offensive" examples and only 706 "Offensive" examples. This significant skew towards the majority class presents a key challenge for model training.
- *Kannada–English:* This corpus consists of 5874 examples. It shows a notable class imbalance, with 4397 "Not-Offensive" examples and only 1477 "Offensive" examples.

4.3 Implementation

The practical pipeline for implementation of FLAICOL follows three main phases: Fine-tuning, Finding Flip points, Augmentation, and retraining.

4.3.1 Fine-tune base classifier

For each corpus we fine-tune a pretrained Transformer sequence-classification model on the training split. The fine-tuned model provides logits $z(\cdot)$ and encoder outputs $h(\cdot)$ required by the flip discovery procedure.

4.3.2 Find flip points

For each example x chosen for analysis:

1. Compute the model prediction $i = \arg\max z(x)$ and select a target class j (typically the highest-scoring alternative).
2. Extract the encoder representation $h(x)$ and solve the bias subproblem (equation (2)) to obtain b_{homo}.
3. Interpolate the bias from b_{homo} back to b_{orig} in several homotopy steps. At each step fix the temporary bias and perform a local embedding-space search:
   - Select a small set of token positions to perturb.
   - Optimize offsets Δ for those positions by minimizing the penalty objective (3) with a gradient-based optimizer (Adam) and a schedule of penalty weights α.
   - If the equality $z_i \approx z_j$ and the inequality conditions hold, treat the solution as successful and warm-start the next homotopy step with the found offsets.
4. After the final homotopy step, map the continuous perturbed embeddings $E_0 + \Delta$ to discrete tokens via nearest-neighbor in the embedding matrix, run the discrete candidate through the original classifier, and keep validated flips that meet the flip criteria.

4.3.3 Augmentation and retraining

Obtained flip texts are converted into synthetic training examples and appended to the original training data. The classifier is retrained on this augmented dataset, and held-out evaluation is used to measure the effect of flip-based augmentation.

As a simple baseline for minority-class augmentation, we also ran SMOTE (synthetic minority oversampling) (Chawla et al., 2002) on the training split and retrained the same classifier architecture for direct comparison with flip-point augmentation, the backbone behind the proposed FLAICOL.

5 Results

We evaluate FLAICOL using two encoder backbones—XLM-RoBERTa-base (Kodali et al., 2025) and MURIL-base (Khanuja et al., 2021) fine-tuned on the Malayalam-English, Tamil-English, and Kannada-English splits of the Dravidian-CodeMix corpus. For each language we generate a certain number ({100, 150, 200 }) of flip-derived examples and append them to the training set. Overall, FLAICOL yields modest but directionally consistent gains in validation performance, with the largest improvements observed for the minority (Offensive) class, suggesting reduced near-boundary errors.

5.1 Distributional plausibility and surface fluency of flips

We assess reconstructions with three complementary, deterministic diagnostics: (i) lexical/phrase diversity measured by Self-BLEU (S-BLEU$_{2/3/4}$, lower = more diverse), (ii) per-pair semantic fidelity measured by pairwise LaBSE cosine (reported as LaBSE$_{\text{mean}}$ and the 90th percentile LaBSE$_{p90}$) and (iii) set-level distributional similarity measured by a Fréchet Embedding Distance (FED) (Alihosseini et al., 2019) computed on LaBSE embeddings (bootstrap mean with 95% CI). Table 1 reports these statistics for the four experimental conditions.

The table shows a clear backbone-level separation. XLM-R reconstructions preserve semantics substantially better than MuRIL, with LaBSE$_{\text{mean}}$ similarity in the mid 0.7 range and upper-tail (p90) values approaching 0.9, whereas MuRIL remains in the mid 0.3 range. At the set level, distributional drift is also consistently smaller for XLM-R (FED $\approx$ 0.43–0.46) than for MuRIL ($\approx$ 0.62–0.63), and the bootstrap confidence intervals do not overlap, suggesting a statistically stable separation across conditions.

Self-BLEU reveals a complementary trade-off. MuRIL reconstructions show much lower Self-BLEU values (around 0.03–0.06), indicating greater lexical diversity, while XLM-R reconstructions fall in a higher range (roughly 0.08–0.28), reflecting greater phrase reuse. Taken together with the LaBSE and FED results, this suggests that MuRIL achieves higher surface diversity at the cost of semantic fidelity and increased distributional shift, whereas XLM-R maintains closer semantic and distributional alignment with slightly lower lexical variation.

These metrics indicate a consistent encoder-level separation in semantic fidelity and distributional alignment.XLM-R reconstructions remain closer to the original semantic manifold, whereas MuRIL reconstructions exhibit greater stylistic and distributional divergence. This difference reflects how encoder representations shape the stability of flip-based generation.

5.2 Qualitative comparison of flip-finding backbones

Representative reconstructions are shown adjacent to their descriptions to emphasize the surface phenomena that underlie the metrics.

5.2.1 XLM-R reconstructions:

XLM reconstructions typically:

- preserve Tamil/Kannada/Malayalam–English code-mixing and script fidelity,
- retain colloquial, comment-style phrasing,
- introduce small, interpretable lexical and spacing perturbations, and
- maintain semantic similarity to the original comments.

In several cases, the reconstructed text is nearly identical to the original, differing only by additional whitespace, underscores, or minor tokenization artifacts. These minimal surface variations explain the lower FED and higher LaBSE observed for XLM flips, decoded token sequences often remain very close to the originals, as reflected by low normalized token-edit distance and relatively higher Self-BLEU, indicating local and controlled edits rather than substantial rewrites.

Table 1: Evaluation of generated flip-points.

Backbone	Language	S-BLEU$_{2/3/4}$	LaBSE$_{mean}$	LaBSE$_{p90}$	FED$_{mean}$ (95% CI)
XLM-R	Mal	0.285 / 0.139 / 0.083	0.726	0.876	0.461 (0.412–0.512)
XLM-R	Tam	0.271 / 0.108 / 0.058	0.759	0.896	0.436 (0.391–0.483)
XLM-R	Kan	0.249 / 0.159 / 0.121	0.762	0.908	0.433 (0.390–0.481)
MuRIL	Mal	0.058 / 0.041 / 0.035	0.360	0.633	0.631 (0.589–0.687)
MuRIL	Tam	0.060 / 0.037 / 0.030	0.373	0.663	0.618 (0.579–0.659)
MuRIL	Kan	0.122 / 0.081 / 0.064	0.384	0.678	0.602 (0.559–0.635)

language	text	reconstructed	translation
Malayalam	കിടുക്കാച്ചി പടം ആയിരിക്കട്ടെ എല്ലാവിധ ആശംസകളും നേരുന്നു	കി ടുക്ക ാ ച്ചി പട ം ആയിര ിക്ക ട്ടെ എല്ലാ വിധ ആശംസ കളും നേര ുന്നു	May it be an awesome movie. All the best.
Malayalam	Mammokka nigalude hardworkinu nalla vijayam thane kittum	Mam mo kka ni gal ude hard work inu na lla vijay am than e kit tum	Mammokka, your hard work will definitely bring great success.
Tamil	எல்லாம் ok.. இந்த உதயநிதி பய தான் சொதபிடுவானோனு பயம்மா இருக்கு.	எல்லாம் ok இந்த உத ய நிதி பய தான் சொ த பி டு வா னோ ந ா பய ம்மா இருக்கு	Everything is good, but I'm scared Udhayanidhi might mess it up
Tamil	NerKondaParvai Thala fans like panna vanam... தல வெறியன்கள் like பண்ணுங்க	N er Kon da Par vai Thala fans like panna van am த ல வெறி யன் கள் like பண்ண ுங்க	Nerkonda Paarvai Thala fans, show your support... Die-hard Thala fans, hit like
Kannada	ಭಾರತದಲ್ಲಿ ಕಂಪನಿ ಸುರು ಮಾಡಲು ಬಂದ್ರೆ ಸಿಟಿ ಬಿಟ್ಟು ಹಳ್ಳಿಯ ಕಡೆಗೆ ಬರೊದೆ ಇಲ್ಲ.ಇಉತ್ತಮ ಮತ್ತು ವಿಶಾಲ ಪ್ರದೇಶ ಇರೊದು ಹಳ್ಳಿಯಲ್ಲಿ .ಮೊದಲು ಅದನ್ನ ಸರ್ಕಾರ ಮಾಡಬೇಕು.	ಭಾರತದಲ್ಲಿ ಕಂಪನಿ ಸು ರು ಮಾಡಲು ಬಂದ್ ರೆ ಸಿ ಟಿ ಬಿಟ್ಟು ಹಳ್ಳಿ ಯ ಕಡೆ ಗೆ ಬರ ೊ ದೆ ಇಲ್ಲ ಇ ಉ ತ್ತ ಮ ಮತ್ತು ವಿ ಶಾ ಲ ಪ್ರದೇಶ ಇರ ೊ ದು ಹಳ್ಳಿ ಯಲ್ಲಿ ಮ ೊ ದ ಲು ಅದ ನ್ನ ಸರ್ಕಾರ ಮಾಡಬೇಕು	When companies come to start in India, they don't look beyond cities. The best and most spacious areas are in villages; the government should prioritize that first.
Kannada	ತುಂಬಾ ಚನ್ನಾಗಿ ಮೂಡಿ ಬಂದಿದೆ ಮತ್ತು ನಿರೀಕ್ಷೆ ಹುಟ್ಟಿಸುವಂತಿದೆ. ಸಿನಿಮಾ ಬಿಡುಗಡೆಗೆ ಕಾಯುತ್ತೇವೆ. ಒಳ್ಳೆಯದಾಗಲಿ	ತುಂಬಾ ಚ ನ್ನ ಾಗಿ ಮೂ ಡಿ ಬಂದಿದೆ ಮತ್ತು ನಿರೀಕ್ಷೆ ಹುಟ್ಟ ಿಸುವ ಂತ ಿದೆ ಸಿನಿಮಾ ಬಿಡುಗಡೆ ಗೆ ಕಾಯ ುತ್ತೇವೆ ಒಳ್ಳೆಯ ದಾಗ ಲಿ	It has come out very well and looks promising. Waiting for the release; all the best!

Figure 2: Examples of XLM-based flip reconstructions.

5.2.2 MuRIL reconstructions:

MuRIL reconstructions frequently:

- include cross-script insertions (e.g., Odia, Bengali, Devanagari fragments),
- contain multi-language segments unrelated to the input context,
- show large structural edits and irregular token boundaries and
- display pronounced semantic drift from the source comment.

These behaviors explain the higher FED and lower LaBSE for MuRIL flips, decoded sequences exhibit larger surface changes and greater lexical novelty, reflected in higher normalized token-edit distance and lower Self-BLEU.

Lower FED together with higher LaBSE similarity and small normalized token-edit distance indicates that XLM reconstructions remain close to the source distribution while preserving semantic structure and surface patterns. In contrast, higher FED combined with lower LaBSE, larger normalized edits, and lower Self-BLEU for MuRIL reconstructions reflects stronger stylistic and distributional divergence, often including noisier or cross-script variations. Practically, these findings suggest that XLM flips are more directly aligned with the original data manifold, whereas MuRIL flips require stricter similarity or structural constraints prior to inclusion.

Table 2: PMI-based analysis for the offensive class across MURIL and XLM settings.

Dataset	Original offensive PMI	Flip offensive PMI
MAL-XLM	4.79%	0.00%
TAM-XLM	0.71%	0.00%
KAN-XLM	2.70%	0.55%
MAL-MuRIL	4.79%	0.00%
TAM-MuRIL	0.71%	0.29%
KAN-MuRIL	2.70%	0.00%

5.2.3 PMI analysis

As shown in Table 2, flip-generated offensive samples exhibit substantially lower overlap with high-

language	text	reconstructed	translation
Malayalam	Apt മ്യൂസിക് Sushin.. Hats off	Apt ସମ୍ପ୍ରତ± ચાવીરૂપ लेखांची 1797 سکینs ପ୍ରତାପ ग्रैगरी	Spot on music, Sushin... Hats off, spot on
Malayalam	രാക്ഷസൻ ഇമൈകൾനോടികൾ ഒക്കെ അങ്ങോട്ട് മാറിനിക്ക് ഇനി ഇവിടെ 5-ആം പാതിര ഭരിക്കും	രാക്ഷസൻ ഇമൈകൾനോടികൾ 6500 അ ডাকনামনിക്ക് क्रीडांगण ഇവിടെ 5 ആം TDुषा ഭ पञ्चाङ्गानुसारं जवळपासची	Step back, Ratsasan and Imaikkaa Nodigal. Anjaam Pathira is the new ruler of this genre
Tamil	BGM asusual therikka vitaru Sam cs	BGM కొత్తవీధి وزیرِ खेलकुदकोu نیاپو cs 139	As usual, Sam CS is just blowing it up with the BGM
Tamil	Waiting for lokesh kanakaraj Thalapathy 64 Thalapathy makkal selvam 2020 summer boxoffice	for lokesh kanakaraj Thalap निकिता 64 Thalap मकैको वहाँको। 600px 2020 summer box 380ice,Waiting for Lokesh Kanagaraj's Thalapathy 64, "Thalapathy and Makkal Selvam are going to own the 2020 Summer box office"	Waiting for Lokesh Kanagaraj's Thalapathy 64. Thalapathy and Makkal Selvam are going to own the 2020 Summer box office
Kannada	ಅದ್ಭುತವಾಗಿದೆ ಅವನೇ ಶ್ರೀಮನ್ನಾರಾಯಣ ಚಿತ್ರದ ಹಾಡು ಹ್ಯಾಂಡ್ಸಪ್	അക്ഷരമാലയിലെദ്ധ്ധ थानि ՐեՐե ಶ್ರೀಮನ್ನಾರಾಯಣ ಬೆನ್ನಿನ বার্বাডোসের കൈതപ്രം ہلپک	Amazing song, 'Hands Up' from Avane Srimannarayana.
Kannada	ಸೂಪರ್ ಸಾಂಗ್ ರಕ್ಷಿತ್ ಶೆಟ್ಟಿ ಸರ್ ನನಮ್ಮ ಉಡುಪಿ ಜಿಲ್ಲೆಯ ಹೆಮ್ಮೆ	ಸೂಪರ್ پاکیزہ ರಕ್ಷಿತ್ नेपालमै बिहाग सीमात शब्दाची માળિયા उपज़िले	Great song! Rakshit Shetty sir is the pride of our Udupi.

Figure 3: Examples of MuRIL-based flip reconstructions.

PMI offensive tokens than the original offensive data across both XLM-R and MuRIL settings. Most configurations show zero overlap, with only minor residual overlap observed for TAM-MuRIL and KAN-XLM. These results suggest that the generated flips are not primarily driven by dominant offensive lexical triggers, indicating that FLAICOL does not merely exploit simple PMI-based shortcuts.

5.2.4 Acceptance Rate

To quantify how often a latent-space flip remains valid after discretization, we measure the acceptance rate, defined as the proportion of perturbations that still produce a label change after projection back to token space. This provides a practical measure of discrete survivability, since a perturbation is only useful for augmentation if the label flip persists after continuous-to-discrete conversion.

MuRIL-based settings achieve acceptance rates of 44% (MAL-MuRIL), 53% (TAM-MuRIL), and 71% (KAN-MuRIL), indicating that a substantial fraction of latent flips survive discretization. In contrast, XLM-R settings produce very few perturbations that satisfy the strict post-projection flip validation criterion, despite frequent success during latent-space optimization. Qualitative inspection shows that many XLM-R reconstructions remain extremely close to the original inputs, often differing only in whitespace or tokenization artifacts. This suggests that the model may already be relatively invariant to such minor surface perturbations, causing many projected candidates to remain within the original decision region after discretization.

5.3 Classification impact of flip-point augmentation

We included 100 flip-based augmentation samples, along with larger augmentation settings of 150 and 200 samples, and evaluated all configurations across three random seeds. The resulting Offensive-F1 scores are summarized in Table 3.

As shown in Table 3, FLAICOL consistently improves Offensive-F1 across all six language–model configurations. For XLM-R, the largest gains are generally observed at 200 augmentation samples, with Malayalam showing the strongest improvement (+3.89% points over baseline). For MuRIL, the best performance is typically obtained at 150 samples, particularly for Kannada (+2.84% points), suggesting that moderate augmentation levels may provide a better balance between diversity and stability. To assess robustness, we performed paired t-tests between the baseline and the best-performing augmentation setting for each configuration (last column of Table 3). Although improvements were consistently observed across seeds, statistical significance was not reached under the current setup, likely due to the modest gains.

Table 4 compares flip-point augmentation with a standard SMOTE baseline (Chawla et al., 2002) (both methods appended 100 synthetic samples). Overall, flip-point consistently matches or slightly outperforms SMOTE , where flip-derived examples produce a clearer uplift in minority-class F1.

Table 3: Offensive-class F1 scores (mean ± standard deviation over 3 random seeds) for augmentation settings with 100, 150, and 200 flip-derived samples. $p^{\dagger}$ denotes the paired t-test p-value between the baseline and the best-performing augmentation setting for each language–model configuration.

Language	**Model**	**Offensive-F1 (Mean ± Std)**				$\mathbf{p}^{\dagger}$
		Baseline	**+100**	**+150**	**+200**	
Malayalam	XLM-R	0.6467 ± 0.0459	0.6534 ± 0.0245	0.6569 ± 0.0427	**0.6856 ± 0.0217**	0.116
Tamil	XLM-R	0.6669 ± 0.0036	0.6684 ± 0.0062	0.6754 ± 0.0078	**0.6830 ± 0.0070**	0.118
Kannada	XLM-R	0.6215 ± 0.0159	0.6233 ± 0.0124	0.6268 ± 0.0183	**0.6374 ± 0.0078**	0.259
Malayalam	MuRIL	0.6477 ± 0.0181	0.6563 ± 0.0542	0.6612 ± 0.0103	**0.6635 ± 0.0215**	0.104
Tamil	MuRIL	0.6831 ± 0.0016	0.6872 ± 0.0173	**0.6881 ± 0.0099**	0.6868 ± 0.0089	0.148
Kannada	MuRIL	0.6435 ± 0.0102	0.6505 ± 0.0061	**0.6719 ± 0.0193**	0.6712 ± 0.0292	0.158

Table 4: Validation F1 (Offensive) — comparison of SMOTE vs FLAICOL

Model	Language	SMOTE	FLAICOL (Proposed)
XLM-R	Malayalam	0.626	**0.641**
XLM-R	Tamil	0.654	**0.664**
XLM-R	Kannada	0.611	**0.615**
MURIL	Malayalam	0.704	**0.726**
MURIL	Tamil	0.669	**0.671**
MURIL	Kannada	0.626	**0.640**

Compared to the non-augmented baseline, only a subset of augmentation runs produce gains: some SMOTE configurations did not outperform the base model, whereas flip-point more reliably matched or exceeded baseline performance. This pattern suggests (i) flip samples are most helpful when they introduce meaning-preserving, targeted variations near decision boundaries, and (ii) when generated samples are largely redundant with existing data their incremental benefit is smaller.

XLM flips produce reconstructions that are distributionally closer to the original instances and more semantically faithful, aligning well with the original data manifold. MuRIL flips exhibit greater stylistic variation and distributional shifts and therefore may benefit from stronger similarity or structural constraints before inclusion. Although absolute metric values depend on embedding choice and preprocessing configuration, the relative separation between XLM and MuRIL remains consistent across diagnostics.

6 Summary

Hate-speech detection in code-mixed social media text remains challenging due to script variation, language mixing, and scarce annotated data. We developed FLAICOL, a practical flip-point pipeline for pretrained Transformer classifiers that traces a homotopy-based path to produce small, targeted token-level perturbations and converts continuous solutions into discrete augmentation examples. Applied to the Malayalam–English, Tamil–English and Kannada-English splits of DravidianCodeMix, appending a small set of flips yields measurable robustness gains; overall validation performance improves (macro-F1 increases by roughly one percentage point on our primary runs), and minority-class performance improves modestly.

Beyond these overall gains, our analysis reveals two consistent findings. Reconstructions produced by the XLM encoder remain distributionally and semantically closer to the original corpus, as indicated by lower Fréchet Embedding distance and higher embedding-based similarity, remaining closer to the original embedding distribution. MuRIL reconstructions exhibit greater stylistic divergence and larger distributional shift, reflected in higher Fréchet Embedding distance, lower semantic similarity, and increased lexical variation. These patterns indicate that the choice of encoder meaningfully shapes the semantic stability and distributional realism of flip-generated samples.

FLAICOL combines interpretability with data efficiency; a small set of boundary-targeted flips can strengthen near-decision examples and yield modest but consistent improvements in low-resource, script-diverse, code-mixed settings.

7 Practical implications and Limitations

XLM-style flip generation yields reconstructions that are distributionally closer, more fluent, and more reliable for augmentation in code-mixed Tamil, Kannada and Malayalam data. MuRIL-style reconstructions, while capable of yielding

classification gains, introduce substantial stylistic noise and therefore benefit from stronger filtering (e.g., script constraints or stricter similarity thresholds). Absolute metric values depend on the chosen embedding model (LaBSE) and language-model configuration. However, the relative separation between XLM and MuRIL observed here is consistent across our evaluations. Flip points help us to understand how the model makes decisions about its prediction boundaries. Specifically, if small, non-semantic changes to an input result in a change to the predicted label, it shows that the classifier is dependent on shallow or non-semantic signals and not on robust semantic signals. From this perspective, FLAICOL is a diagnostic tool to show weaknesses in representations that have been learned from the dataset or to indicate that a classifier may rely on artifacts created by the dataset they were trained with or token-level biases. This also fits the larger goal of making models more reliable, particularly in low-resource and noisy code-mixed contexts. Although FLAICOL consistently improves Offensive-F1 across configurations, statistical significance was not reached under the current experimental setup. This is likely due to the modest absolute gains and the limited number of random seeds evaluated. While the experiments in this paper use Dravidian CodeMix benchmark, future work can extend this to other benchmarks. In particular, it would be interesting to examine the Bengali-Hindi-English trilingual code-mixed corpus curated in recent efforts, which, although slightly falling outside the scope of this workshop focusing on NLP techniques in Dravidian languages, shall be of interest to the broader low-resource NLP research community. Future work shall also explore combining flip-based augmentation with generative methods such as back-translation.

Acknowledgements

The authors place their heartfelt gratitude to the resources provided by the institute to which they are affiliated and their department, which is sponsored by the prestigious DST-FIST Initiative of the Government of India. We sincerely thank the reviewers whose constructive feedback has helped shape the camera-ready draft of the paper.

References

Abdullah Al Nahian, Mst Rafia Islam, Azmine Toushik Wasi, and Md Manjurul Ahsan. 2025. Nlpopsciol@dravidianlangtech 2025: Classification of abusive tamil and malayalam text targeting women using pre-trained models. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 38–45.

Danial Alihosseini, Ehsan Montahaei, and Mahdieh Soleymani Baghshah. 2019. Jointly measuring diversity and quality in text generation models. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 90–98, Minneapolis, Minnesota. Association for Computational Linguistics.

S Anbukkarasi and S Varadhaganapathy. 2023. Deep learning-based hate speech detection in code-mixed tamil text. *IETE Journal of Research*, 69(11):7893–7898.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John P McCrae. 2022. Dravidiancodemix: Sentiment analysis and offensive language identification dataset for dravidian languages in code-mixed text. *Language Resources and Evaluation*, 56(3):765–806.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Seffi Cohen, Dan Presil, Or Katz, Ofir Arbili, Shvat Messica, and Lior Rokach. 2023. Enhancing social network hate detection using back translation and gpt-3 augmentations during training and test-time. *Information Fusion*, 99:101887.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Subhajeet Das, Koushikk Bhattacharyya, and Sonali Sarkar. 2023. Performance analysis of logistic regression, naive bayes, knn, decision tree, random forest and svm on hate speech detection from twitter. *International Research Journal of Innovations in Engineering and Technology*, 7(3):24.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, volume 1 (long and short papers)*, pages 4171–4186.

Koyel Ghosh and Apurbalal Senapati. 2022. Hate speech detection: a comparison of mono and multilingual transformer model with cross-language evalu-

ation. In *Proceedings of the 36th Pacific Asia Conference on Language, Information and Computation*, pages 853–865.

Dhiman Goswami, Nishat Raihan, Antara Mahmud, Antonios Anastasopoulos, and Marcos Zampieri. 2023. Offmix-3l: A novel code-mixed test dataset in bangla-english-hindi for offensive language identification. In *Proceedings of the 11th International Workshop on Natural Language Processing for Social Media*, pages 21–27.

Shenson Joseph and Herat Joshi. 2024. Ulmfit: universal language model fine-tuning for text classification. *International Journal of Advanced Medical Sciences and Technology*, 4(6):10–54105.

Pallabi Kakati and Devendra Dandotiya. 2024. Automatic detection of hate speech in code-mixed indian languages in twitter social media interaction using dconvblstm-muril ensemble method. *Social Network Analysis and Mining*, 14(1):108.

Vidhya Kamakshi and Narayanan C Krishnan. 2023. Explainable image classification: The journey so far and the road ahead. *AI*, 4(3):620–651.

Sanjana Kavatagi and Rashmi Rachh. 2025. Hastika: hate speech and target identification in kannada-english code-mixed text. *Language Resources and Evaluation*, pages 1–46.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, and 1 others. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Diederik P Kingma and Jimmy L Ba. 2015. Adam : A method for stochastic optimization. *International Conference on Learning Representations*, 7.

Rohith Gowtham Kodali, Durga Prasad Manukonda, and Daniel Iglesias. 2025. bytesizedllm@ nlu of devanagari script languages 2025: Hate speech detection and target identification using customized attention bilstm and xlm-roberta base embeddings. In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025)*, pages 242–247.

Niranjan Kumar, Pranav Gupta, Billodal Roy, and Souvik Bhattacharyya. 2025. Lexilogic@ dravidian-langtech 2025: Detecting misogynistic memes and abusive tamil and malayalam text targeting women on social media. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 435–439.

Ritesh Kumar, Bornini Lahiri, Atul Kr Ojha, and Akanksha Bansal. 2020. Comma@ fire 2020: Exploring multilingual joint training across different classification tasks. In *FIRE (Working Notes)*, pages 823–828.

Jiaxiang Liu, Xuyi Chen, Shikun Feng, Shuohuan Wang, Xuan Ouyang, Yu Sun, Zhengjie Huang, and Weiyue Su. 2020. Kk2018 at SemEval-2020 task 9: Adversarial training for code-mixing sentiment classification. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 817–823, Barcelona (online). International Committee for Computational Linguistics.

Endang Wahyu Pamungkas and Viviana Patti. 2019. Cross-domain and cross-lingual abusive language detection: A hybrid approach with deep learning and a multilingual lexicon. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 363–370.

Md Nishat Raihan, Dhiman Goswami, Antara Mahmud, Antonios Anastasopoulos, and Marcos Zampieri. 2023. Sentmix-3l: A bangla-english-hindi code-mixed dataset for sentiment analysis. In *Proceedings of the First Workshop in South East Asian Language Processing*, pages 79–84.

Nishat Raihan, Dhiman Goswami, Antara Mahmud, Antonios Anastasopoulos, and Marcos Zampieri. 2024. Emomix-3l: a code-mixed dataset for bangla-english-hindi for emotion detection. In *Proceedings of the 7th Workshop on Indian Language Data: Resources and Evaluation*, pages 11–16.

Priya Rani, Shardul Suryawanshi, Koustava Goswami, Bharathi Raja Chakravarthi, Theodorus Fransen, and John Philip McCrae. 2020. A comparative study of different state-of-the-art hate speech detection methods in hindi-english code-mixed data. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 42–48.

Pradeep Kumar Roy and Abhinav Kumar. 2025. Ensuring safety in digital spaces: Detecting code-mixed hate speech in social media posts. *Data & Knowledge Engineering*, page 102409.

V Sanh. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *Proceedings of Thirty-third Conference on Neural Information Processing Systems (NIPS2019)*.

Happy Khairunnisa Sariyanto, Diclehan Ulucan, Oguzhan Ulucan, and Marc Ebner. 2025. Towards explainable hate speech detection. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12883–12893.

Sunil Saumya, Abhinav Kumar, and Jyoti Prakash Singh. 2021. Offensive language identification in dravidian code mixed social media text. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 36–45.

Rajat Sharma, Nikhil Reddy, Vidhya Kamakshi, Narayanan C Krishnan, and Shweta Jain. 2021. Maire-a model-agnostic interpretable rule extraction procedure for explaining classifiers. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 329–349. Springer.

Rizwana Kallooravi Thandil, Muneer Vk, and Sabique Pv. 2025. A novel deep learning framework with advanced feature engineering for hate speech detection in accented malayalam speech. *Humanities and Social Sciences Communications*, 13(1):10.

P Deepasree Varma, P Vinod, M Nandakumar, K Akshay, and Akhil Madhu. 2022. Hate speech detection in english and malayalam code-mixed text using bert embedding. In *2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS)*, pages 1–6.

Roozbeh Yousefzadeh and Dianne P. O'Leary. 2020. Deep learning interpretation: Flip points and homotopy methods. In *Proceedings of The First Mathematical and Scientific Machine Learning Conference*, volume 107 of *Proceedings of Machine Learning Research*, pages 1–26. PMLR.