Gujarati Hate Speech Classifier — Capstone Report

Active learning with IndicBERT fine-tuning, pseudo-labeling, and class-weighted training

Author: Parva Shah | Email: ps7384@rit.edu | Date: August 12, 2025

Abstract

I built a binary text classifier that detects hate speech in Gujarati. The system fine-tunes ai4bharat/indic-bert on a human-labeled seed set, iteratively expands coverage with high-confidence pseudo-labels from a large unlabeled pool, and retrains with class-weighted loss to counter label imbalance. On a held-out validation set, the best checkpoint achieves ~85% accuracy with F1≈0.88 for the hate class.

1. Dataset

All CSVs use schema text,label with labels: 0=non-hate, 1=hate. The unlabeled pool has only text.

1.1 Corpus summary (current project snapshot)

Split / File	Rows	Class 0	Class 1	Notes
seed_labels.csv	398	199	199	Initial human-labeled seed set
expanded_seed.csv	598	363	235	Additional curated labels
uncertain_labels.csv	200	164	36	Manually resolved ambiguous/edge cases
pseudo_labels.csv	50	0	50	High-confidence model predictions (thresholded)
unlabeled_pool.csv	65,158	—	—	Gujarati youtube comments from mining

Source text consists primarily of short, informal Gujarati comments that often include code-mixed English, slang, and spelling variation—all common failure modes for off-the-shelf models without targeted fine-tuning.

2. Method

2.1 Base model

I fine-tuned ai4bharat/indic-bert with a linear classification head. Tokenization uses the model's native WordPiece/BPE vocabulary. Maximum sequence length is 128 tokens.

2.2 Training setup

Phase	Epochs	Batch size	Learning rate	Max length	Other
Initial (gold-only)	5	25	2e-5	128	Stratified hold-out; eval each epoch
With pseudo (gold+pseudo)	4	25	2e-5	128	Resume from initial checkpoint

2.3 Class imbalance handling

I computed inverse-frequency class weights on the training mix and pass them to a weighted cross-entropy loss (custom WeightedTrainer wrapper). This stabilizes learning when non-hate examples dominate.

2.4 Pseudo-labeling

After the initial fine-tune, I scored a sampled subset of the unlabeled pool and keep predictions above a confidence threshold of 0.80. If too few items pass, I backfilled to a minimum of TOP_K_MIN = 50. To curb confirmation bias, I capped the amount of pseudo data mixed into training to approximately a 1:1 ratio with gold labels.

2.5 Active-learning loop

The workflow supports repeating: re-generate pseudo labels using the latest checkpoint, retrain with gold+pseudo, and monitor validation. I also maintain a small "uncertain" bucket for manual review (edge cases, sarcasm, code-mixed slang).

3. Experiments & Results

Metric	Value
Validation Accuracy	0.853
F1 (hate=1)	0.878
Precision (hate=1)	0.818
Recall (hate=1)	0.947
Validation Loss	0.532

Metrics are reported on a held-out split drawn from gold labels only. The model prioritizes recall for the hate class, which is desirable for flagging harmful content; threshold sweeps can trade recall for precision depending on deployment needs.

3.1 Observations

• Adding a modest number of high-confidence pseudo labels improved recall without collapsing precision.
• Class-weighted loss reduced the tendency to predict the majority class on mixed, imbalanced batches.

• Most residual errors occur on short, sarcastic comments, heavy code-mixing, or creative spellings.

4. Limitations & Ethics

• Domain shift: Social-platform slang changes quickly; periodic refresh and re-labeling is required.
• Bias: Even with balanced seeds, sampling and annotation bias can creep in. I avoid demographic attributes as features and report class-wise metrics.
• Ambiguity: Sarcasm and quoted speech remain challenging without richer context.

5. Future Work

• Expand gold labels via assisted annotation (active learning; uncertainty sampling; disagreement mining).
• Upgrade backbone (IndicBERT v2, MuRIL, or XLM-R) and compare with parameter-efficient adapters (LoRA, IA3).
• Character/phoneme-aware augmentation to handle spelling variants and code-mixed forms.
• Calibrated decision thresholds per domain; add a third class for abusive but non-hate content.
• Robustness and fairness checks (stratify by topic, slang type; adversarial rephrasing tests).
• Lightweight inference (ONNX export) for real-time moderation.

6. References

1. AI4Bharat — IndicBERT model card and resources.
HuggingFace Transformers & Datasets documentation.
Relevant literature on Indic hate-speech detection and code-mixed NLP.