# Our Research, Explained Simply

Quantifying and Mitigating Bias in Hindi/Hinglish Toxicity Classification

Using the Feynman Technique — if you can't explain it simply, you don't understand it well enough.

## 1. What's the Problem?

Social media platforms use computer programs to automatically detect hateful or abusive posts. These programs are called **toxicity classifiers** — they read a piece of text and decide whether it's toxic (hateful, abusive) or safe.

> **Toxicity Classifier** — A computer program that reads text and predicts whether it is hateful/abusive (toxic) or not.

Here's the catch: these programs sometimes cheat. Instead of understanding *what* a sentence actually means, they learn shortcuts. If the training data has a lot of hateful posts that mention the word "Muslim" or "Dalit," the program starts thinking that just *mentioning* these words means a post is toxic — even when it isn't. A perfectly normal sentence like "My friend is Muslim" might get flagged as hate speech.

> *Think of it this way: Imagine a teacher who notices that students who wear red shoes tend to get bad grades. Instead of checking the actual homework, the teacher just starts giving bad grades to everyone wearing red shoes. That's what these classifiers are doing — punishing the "shoes" (identity words) instead of reading the "homework" (actual meaning).*

This is a big deal because it means people from certain communities — already marginalised in many cases — get their posts removed more often, their voices silenced more often. The system that's supposed to protect everyone ends up being unfair to the people who need protection the most.

## 2. Why Hindi and Hinglish?

Almost all the research on this fairness problem has been done in English. But hundreds of millions of people in India speak Hindi or Hinglish — a mix of Hindi and English that's extremely common on Indian social media.

> **Hinglish (Code-Mixing)** — Writing or speaking using a mix of Hindi and English in the same sentence. Example: "Yeh movie bohot acchi thi, totally loved it."

In India, online conversations about religion, caste, and politics are intense and heavily moderated. Nobody had checked whether toxicity classifiers for Hindi/Hinglish have the same bias problem. We decided to check. Spoiler: they do.

## 3. Our Data

We used a publicly available dataset of 29,550 Hindi/Hinglish social media posts from Kaggle, each labelled as either hate or non-hate. We split it into three parts: a training set (70%) to teach the model, a development set (10%) to tune it, and a test set (20%) to evaluate it on text it has never seen before.

| 29,550 | 46.4% | 53.6% |
|---|---|---|
| Total Posts | Hateful | Non-Hateful |

We also built an **identity lexicon** — a list of words related to four types of identity in India: religion (Hindu, Muslim, Sikh...), caste (Dalit, Brahmin...), gender (woman, man, ladki...), and region (Bihari, Madrasi...). We tagged every post that contained any of these words so we could track how the model treats each group.

> **Identity Lexicon** — A curated list of words that refer to specific social groups. We use it to tag posts so we can measure bias per group.

## 4. The Model We Tested

We used a model called **XLM-RoBERTa**. Think of it as a very advanced language brain that has been pre-trained by reading massive amounts of text in 100 different languages. It already "understands" many languages, including Hindi and English. We took this pre-trained brain and taught it one specific task: decide if a Hindi/Hinglish post is toxic or not. This process is called **fine-tuning**.

> **Fine-tuning** — Taking a pre-trained model that already knows a lot about language, and training it further on a specific task (in our case, toxicity detection).

> *Think of it this way: It's like hiring someone who already speaks 100 languages and then training them specifically to be a content moderator for Hindi social media.*

## 5. How We Measured Bias

We used three different rulers to measure bias, because one measurement is never enough to see the full picture.

**Ruler 1: False Positive Rate (FPR) Disparity.** A false positive is when the model says a post is toxic when it actually isn't — a false alarm. We checked: does the model produce more false alarms for posts mentioning caste than posts mentioning gender? If the gap between the worst group and the best group is big, the model is biased.

> **False Positive Rate Disparity** — The difference in false alarm rates between the most-flagged identity group and the least-flagged group. Ideally this should be zero.

**Ruler 2: Demographic Parity Difference (ΔDP).** This one is simpler. Regardless of whether the posts are actually toxic or not, does the model predict "toxic" more often for some groups than others? A big gap means the model isn't treating groups equally.

**Ruler 3: Counterfactual Fairness Gap (CFT Gap).** This is the cleverest test. We take a sentence and swap just the identity word — change "Hindu" to "Muslim" and nothing else. If the model is fair, its prediction shouldn't change. If it does change, the model is making decisions based on the identity word, not the actual content.

> **Counterfactual Fairness** — A model is counterfactually fair if changing only the identity word in a sentence doesn't change the model's prediction. We measure this by swapping identity terms and checking how much the prediction shifts.

*Think of it this way: Imagine you submit the same job application twice, identical in every way except the name at the top. If one gets rejected and the other gets accepted, the process isn't fair. That's what our counterfactual test checks.*

## 6. What the Baseline Model Got Wrong

The results were clear: the baseline model has a serious bias problem. Non-toxic posts that mention caste terms were incorrectly flagged as toxic 42% of the time. For gender terms, that number was only 15%. That means the model is nearly **three times more likely** to wrongly censor a post about caste than a post about gender.

| 42.1% | 28.3% | 15.1% |
|---|---|---|
| Caste FPR | Religion FPR | Gender FPR |

**Nearly half of all innocent posts mentioning caste were wrongly flagged as hate speech.**

## 7. Two Ways We Tried to Fix It

We tested two different strategies to reduce this bias. Both leave the core model the same but change how it learns.

**Fix 1: Counterfactual Data Augmentation (CDA).** The idea is dead simple. If the training data has a toxic post that says "Those Muslims are criminals," we create a copy that says "Those Hindus are criminals" — same toxicity label, different identity word. We do this for every identity term across all four categories. Then we train the model on both the original and the swapped data.

> **Counterfactual Data Augmentation (CDA)** — Creating extra training examples by swapping identity words while keeping the label the same. This teaches the model that toxicity depends on what's being said, not who it's about.

> *Think of it this way: It's like showing a child that calling anyone a criminal is mean — not just when you say it about one particular group. You teach fairness by example.*

**Fix 2: Adversarial Debiasing.** This one is more complex. We add a second "brain" to the model — an adversary — whose job is to figure out which identity group a post is about, using only the main model's internal representation. Then we use a trick called a **Gradient Reversal Layer** to train the main model to make the adversary's job *impossible*. If the adversary can't tell which group a post mentions, the main model must have stopped encoding identity information.

> **Gradient Reversal Layer (GRL)** — A mathematical trick used during training. It flips the direction of learning signals from the adversary, so the main model actively learns to *hide* identity information from it.

> *Think of it this way: Imagine two students: one is trying to write essays, the other is trying to guess the writer's religion from the essay. We tell the writer: make your essays so neutral that the guesser can never figure out your religion. Over time, the essays become genuinely identity-neutral.*

## 8. The Scoreboard

Here's how all three models compared. The key insight is in the rightmost columns — the fairness metrics. Lower is better for all three fairness scores.

| | Baseline | CDA | Adversarial |
|---|---|---|---|
| **F1 Score** | 0.703 | **0.685** | 0.699 |
| **FPR Disparity** | 0.421 | **0.196** | 0.256 |
| **DP Gap** | 0.505 | **0.136** | 0.294 |
| **CFT Gap** | 0.052 | **0.042** | 0.068 |

**CDA won on every fairness metric.** It cut the FPR Disparity by 54%, the Demographic Parity gap by 73%, and the Counterfactual Fairness gap by 19%. The cost? Just 1.8 points of F1 score — a tiny price for a much fairer model.

The adversarial model improved on group-level metrics too, but something unexpected happened: it made counterfactual fairness **worse**. The CFT Gap went from 0.052 to 0.068 — a 31% increase. The model got better at looking fair on average, but for any individual sentence, swapping the identity word actually changed the prediction *more* than before.

## 9. The Big Insight: Shallow vs. Deep Fairness

This is the most interesting thing we found. The adversarial approach creates what we call **shallow decorrelation**. On the surface, the statistics look better — the model seems fairer when you look at group averages. But if you zoom into any single sentence and swap an identity word, the model's prediction changes *more* than it did before. The model learned to hide identity information from the adversary without actually stopping its use of identity information when making decisions.

> **Shallow Decorrelation** — When a model appears fair in aggregate statistics but still makes identity-sensitive decisions on individual inputs. The bias is masked, not fixed.

> *Think of it this way: It's like a company that hires equal numbers from every community (looks fair in the statistics) but still treats individual employees differently based on their background once they're inside. The numbers look good; the reality doesn't.*

CDA doesn't have this problem because it fixes the issue at the source — the training data. The model literally sees that the same sentence is toxic regardless of which identity word appears, so it learns not to rely on those words. It's a deeper, more honest kind of fairness.

## 10. Why This Matters

Content moderation systems are used by every major social media platform. When these systems are biased, they disproportionately silence marginalised communities — the very people who are already underrepresented online. In the Indian context, where discussions about caste, religion, and politics are central to public life, unfair moderation can have real consequences: suppressing legitimate discourse, reinforcing social hierarchies, and eroding trust in platforms.

Our work shows that this problem exists in Hindi/Hinglish classifiers, that it's measurable, and that it's fixable — with the right method. CDA is simple, effective, and costs almost nothing in accuracy. There's no good reason not to use it.

## 11. What We Couldn't Do (Honest Limitations)

We used just one dataset. We'd have liked to test on more, but access to some datasets (like HASOC) was never granted. Our identity detection uses a word list, which misses slang, sarcasm, and indirect references. The dataset uses binary labels (hate vs. not-hate) with no distinction between, say, actual hate speech and crude jokes. All our results come from a single random seed — running the experiments multiple times would make the findings more robust. And the region subgroup had only 63 samples, which is too few to draw any real conclusions about regional bias.

---

*"If you can't explain it to a six-year-old, you don't understand it yourself."*

— Often attributed to Richard Feynman

Gupta, Singh, Chandra, Mittal & Nath — SRM Institute of Science and Technology