Multilingual Toxic Comment Detection: Translated vs. Original-Language Training with *Detoxify*

CIS 530 Term Project Proposal

Team: *Zixuan Bian, Aria Shi, Siyuan Shen, Alex Yang*

October 6, 2025

Motivation. Online platforms must moderate harmful content in many languages, yet toxicity detectors trained only on English often fail cross-lingually. The *Jigsaw Multilingual* challenge foregrounded this by evaluating models on non-English comments while most training resources were English [5]. We propose a focused, reproducible study comparing two practical strategies for multilingual toxicity: (i) training on **machine-translated (MT)** variants of English data and (ii) training on **original-language** labeled comments. This comparison matters for deployments balancing accuracy, fairness, and data coverage; *Detoxify* provides public multilingual weights and training scripts for a strong baseline [6], and the 2019 unintended-bias work provides nuanced group metrics we can adopt later if needed [1].

What We Plan to Do

Problem and data (plan paragraph 1). We target **binary toxicity** (score in $[0, 1]$; thresholdable label) following the 2020 multilingual setup [5]. Concretely:

- Use **Jigsaw 2018/2019 (English)** as core sources [3, 4].
- Use **Jigsaw 2020 Multilingual** labeled `validation.csv` to construct per-language splits (e.g., ES/IT/TR; 80/10/10), reserving a held-out test slice for each language [5].
- For the **MT regime**, leverage widely used machine-translated variants of the English corpus referenced during the 2020 competition; for the **original-language regime**, fine-tune directly on labeled non-English splits.

We will report ROC-AUC (primary) and F1 (secondary) *per language* and macro-average across languages.

Models, baselines, and evaluation (plan paragraph 2). Our strong baseline is **Detoxify** (`unitaryai/detoxify`), which provides training scripts and multilingual checkpoints based on XLM-RoBERTa [6, 2]. We will first **reproduce** Detoxify's multilingual results on our splits. Then we implement two extensions: **Extension A (Translated)**—fine-tune with MT corpora; **Extension B (Original-language)**—fine-tune with in-language labeled data. As a **simple baseline**, we include a TF-IDF + Logistic Regression classifier per language. We provide a unified `score.py` that takes predictions + gold labels and outputs ROC-AUC/F1, plus a short error analysis (e.g., profanity-free toxicity, spelling noise) for representative languages.

ES (toxic): "Vete de aquí, nadie quiere leerte." → desired score ≈ 0.9
IT (non-toxic, identity mention): "Sono gay e orgoglioso." → desired score ≈ 0.1
Illustrative examples for the presentation: a toxic insult vs. a neutral identity statement.

Figure 1: Example inputs and desired model behavior.

Inputs and Outputs

Input: raw comment text in language $\ell \in \{\text{ES, IT, TR, } \dots\}$. **Output:** `toxicity` score $\in [0, 1]$ (and optionally a binary label via a tuned threshold). We retain language tags for language-wise metrics.

Data Sources & Code We Will Use

- **Jigsaw datasets (Kaggle CLI).**

```
kaggle competitions download -c jigsaw-toxic-comment-classification-challenge   # 2018
kaggle competitions download -c jigsaw-unintended-bias-in-toxicity-classification  # 2019
kaggle competitions download -c jigsaw-multilingual-toxic-comment-classification   # 2020
```

- **Detoxify (strong baseline + scripts)**: `https://github.com/unitaryai/detoxify` [6]

Planned Baselines and Extensions

- **Simple baseline:** TF-IDF + Logistic Regression (one-vs-rest), trained per language.
- **Strong baseline (reproduction):** Detoxify multilingual (XLM-R) on our splits [6, 2].
- **Extension A (Translated):** fine-tune using machine-translated English→target-language data; evaluate per language.
- **Extension B (Original-language):** fine-tune using original-language labeled data from Jigsaw 2020; evaluate per language [5].

Evaluation and Deliverables

Metrics: ROC-AUC (primary), F1 (secondary), per-language + macro average. **Artifacts:** `score.py`, `simple-baseline.py`, Detoxify training configs, and a short `README` with exact run commands. **Error analysis:** 6–10 annotated cases per language.

References

[1] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion Proceedings of The 2019 World Wide Web Conference*. ACM, 2019. doi: 10.1145/3308560.3317593. URL `https://arxiv.org/abs/1903.04561`.

[2] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020. URL `https://arxiv.org/abs/1911.02116`.

[3] Kaggle and Jigsaw/Conversation AI. Toxic comment classification challenge. `https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge`, 2018. Accessed 2025-10-06.

[4] Kaggle and Jigsaw/Conversation AI. Jigsaw unintended bias in toxicity classification. `https://www.kaggle.com/competitions/jigsaw-unintended-bias-in-toxicity-classification`, 2019. Accessed 2025-10-06.

[5] Kaggle and Jigsaw/Conversation AI. Jigsaw multilingual toxic comment classification. `https://www.kaggle.com/competitions/jigsaw-multilingual-toxic-comment-classification`, 2020. Accessed 2025-10-06.

[6] Unitary AI. Detoxify: Trained models & code for toxic comment classification. `https://github.com/unitaryai/detoxify`, 2025. GitHub repository; accessed 2025-10-06.