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

CIS 5300 Milestone 2

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

November 13, 2025

1 Task and Data Overview

Our term project studies *multilingual toxicity detection* on a collection of Jigsaw toxicity datasets. The goal is to classify a comment as *toxic* or *non-toxic* in several languages (English, Spanish, Italian, and Turkish). The Detoxify model we use as a strong baseline was originally introduced for toxicity detection in online comments [2], and our multilingual setting is closely related to prior work on cross-lingual language models such as XLM-RoBERTa [1].

For each language $\ell \in \{\text{en}, \text{es}, \text{it}, \text{tr}\}$, we construct train/validation/test splits from a common pool of annotated comments. All experiments are performed in a unified code base, and all models output a scalar toxicity probability per comment.

2 Evaluation Metrics

We treat toxicity detection as a binary classification task. For a given language and test set, let $y_i \in \{0, 1\}$ denote the gold label for example i, and let $\hat{p}_i \in [0, 1]$ be the model's predicted toxicity probability. Given a threshold τ, we define the binary prediction

$$\hat{y}_i = \mathbf{1}[\hat{p}_i \geq \tau].$$

From the resulting confusion matrix we obtain

$$TP, FP, TN, FN \in \mathbb{N},$$

the usual numbers of true positives, false positives, true negatives, and false negatives.

All metrics are implemented using the `scikit-learn` library [3].

2.1 ROC–AUC (primary metric)

Our primary metric is the area under the receiver operating characteristic curve (ROC–AUC). The ROC curve plots

$$\text{TPR}(\tau) = \frac{TP(\tau)}{TP(\tau) + FN(\tau)}, \qquad \text{FPR}(\tau) = \frac{FP(\tau)}{FP(\tau) + TN(\tau)}$$

for all thresholds $\tau \in [0, 1]$. ROC–AUC summarizes this curve into a single number between 0 and 1. Intuitively, it is the probability that a randomly chosen toxic comment receives a higher score than a randomly chosen non-toxic comment. We use the implementation in `sklearn.metrics.roc_auc_score`.

2.2 Precision, Recall, and F1 (secondary metrics)

Given a fixed threshold τ and the induced predictions $\hat{y}_i$, we compute

$$\text{Precision} = \frac{TP}{TP + FP}, \qquad \text{Recall} = \frac{TP}{TP + FN},$$

and the F1 score

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

We also report accuracy, specificity, and false positive/negative rates for error analysis, but ROC–AUC and F1 are the main metrics we use to compare models.

2.3 Macro and Micro Averaging Across Languages

For each language ℓ we compute a scalar metric m_ℓ (e.g., ROC–AUC or F1). The macro average over languages is

$$m_{\text{macro}} = \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} m_\ell,$$

which treats all languages equally. For the micro average, we concatenate all predictions across languages and recompute the metrics on this combined set, which weights languages by their number of examples.

3 Baselines

3.1 Simple Baseline: TF–IDF + Logistic Regression

Our simple baseline is a strong traditional model: a per-language TF–IDF + Logistic Regression classifier. For each language, we:

- preprocess the text (lowercasing, URL and mention removal, optional stopword removal);
- vectorize comments with a word-level TF–IDF model (1–2 grams, up to 200k features);
- train a Logistic Regression classifier with balanced class weights and `max_iter = 2000`;
- tune a decision threshold on the validation set to maximize F1.

At test time, the model outputs a probability $\hat{p}_i$ and a binarized prediction $\hat{y}_i$ for each comment. Table 1 summarizes the test-set performance.

Language	ROC-AUC	F1	Precision	Recall	Accuracy
en	0.9686	0.7371	0.7339	0.7404	0.9495
es	0.8604	0.6067	0.5745	0.6429	0.8606
it	0.8430	0.5192	0.4909	0.5510	0.8008
tr	0.9569	0.7368	0.8400	0.6562	0.9500
macro	0.9072	0.6500	0.6598	0.6476	0.8902

Table 1: TF–IDF + Logistic Regression baseline on the test set.

The baseline achieves strong ROC–AUC on English and Turkish, with reasonably high F1. Performance on Spanish and Italian is lower, reflecting smaller dataset sizes and potential domain shift, and leaves room for improvement with stronger models.

3.2 Strong Baseline: Detoxify Multilingual XLM–R

Our strong baseline is the multilingual Detoxify model [2], which uses an XLM–RoBERTa transformer [1] to produce toxicity scores. We load the pretrained `Detoxify("multilingual")` checkpoint and apply it to our processed validation and test splits in each language.

For each language, we:

- run Detoxify to obtain a toxicity probability $\hat{p}_i$ for each comment;
- tune a threshold on the validation set to maximize F1;
- evaluate on the test set using the same ROC–AUC and F1 metrics as above.

Table 2 reports the resulting performance.

Language	ROC-AUC	F1	Precision	Recall	Accuracy
en	0.9889	0.8238	0.8244	0.8232	0.9663
es	0.9184	0.6154	0.8696	0.4762	0.9004
it	0.8756	0.5584	0.4095	0.8776	0.7291
tr	0.9824	0.8219	0.7317	0.9375	0.9567
macro	0.9413	0.7049	0.7088	0.7786	0.8881

Table 2: Detoxify multilingual XLM–R strong baseline on the test set.

Compared to the TF–IDF baseline, Detoxify consistently improves ROC–AUC in all languages (macro ROC–AUC increases from 0.91 to 0.94) and F1 (macro F1 increases from 0.65 to 0.70). The gains are especially large for English and Turkish. For Spanish, Detoxify achieves much higher precision but somewhat lower recall, while for Italian it trades precision for very high recall. Overall, the transformer-based strong baseline provides a substantially stronger starting point for future model improvements.

References

[1] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Armand Joulin, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8440–8451, 2020.

[2] Laura Hanu and Unitary team. Detoxifying language models risks marginalizing minority voices. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP) — Demo Track*, 2021. Detoxify library and models. GitHub repository: `https://github.com/unitaryai/detoxify`.

[3] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.