Report: Toxic Comment Classification (Part 2)

Anastasia Vysotskaya

1 From Part 1 to Part 2

Part 1 of this project implemented classical machine learning models using TF–IDF features. The best model, **Logistic Regression**, achieved strong baseline performance (see Table 1).

Its strengths are *interpretability, speed, and efficiency*, which make it well-suited for large-scale text classification with limited resources. However, its main limitation is that TF–IDF does not capture semantic context or word order, which reduces its ability to recognise subtle or implicit toxicity.

As modern NLP increasingly relies on contextual embeddings, this project (Part 2) extends the previous work with more advanced model types.

2 Word Embeddings

This approach replaces sparse TF–IDF features with dense pretrained representations. Two types of embeddings were tested: **word-level** (fastText vectors) and **sentence-level** (MiniLM). Each embedding set was followed by a Logistic Regression classifier.

Results in Table 1 show that both variants underperformed the TF–IDF baseline. Precision and recall both dropped compared to the classical model, and overall macro F1 remained below the baseline.

3 Transformers

This section employed a transformer model, **DistilBERT**, fine-tuned via Hugging Face pipelines. Despite being a lightweight version, it surpassed all previous approaches, achieving the highest macro F1 and PR–AUC (Table 1).

Transformers capture contextual dependencies and subword semantics, making them highly effective for nuanced toxicity detection. Given more computational resources, a domain-adapted model such as `toxic-bert` would likely yield further improvement.

Thus, while Logistic Regression remains an excellent baseline, a transformer model would be the preferred choice if training time and hardware permit.

4 Prompt Engineering

Prompt-based classification was explored using **Ollama** with the `phi3:mini` model. Three prompt types were tested: zero-shot, few-shot, and role-based moderation prompts. Due to high computational cost, evaluation was limited to a sample of 500 comments.

Even on this subset, results were unstable and showed a high rate of false positives, as reflected by the low precision and PR–AUC in Table 1. Prompt inference proved far slower than supervised models, with runtimes exceeding 30 minutes per run.

Overall, while prompts provide flexibility and require no training data, they are currently impractical for large-scale or time-sensitive moderation tasks.

5 Evaluation

Table 1 summarises the performance of all approaches across main metrics.

Under constrained resources, **Logistic Regression** remains the most balanced choice. It is fast, interpretable, and reliable. The **Transformer model** demonstrated superior performance and would be preferred where computational power allows. **Embedding-based models** did not outperform the baseline, and **prompt methods** proved computationally expensive and less accurate.

In conclusion, contextual models clearly outperform classical features, but cost and deployment complexity remain decisive factors in choosing the optimal approach.

Approach	Precision(1)	Recall(1)	macro F1	ROC–AUC	PR–AUC	Train time (s)	Eval time (s)
Logistic Regression (baseline)	0.8596	0.7371	**0.8868**	0.9786	**0.8872**	–	–
Embeddings (sentence, logreg)	0.8259	0.6308	**0.8445**	0.9608	**0.8203**	12	1
Embeddings (word, logreg)	0.8479	0.6077	**0.8409**	0.9549	**0.8117**	11	1
Transformer (distilbert-base-uncased)	0.8390	0.8156	**0.9046**	0.9855	**0.9132**	2960	337
Prompts (phi3:mini, role, sample of 500)	0.4598	0.8333	**0.7645**	0.8892	**0.5853**	–	2038
Prompts (phi3:mini, zero shot, sample of 500)	0.2979	0.8750	**0.6575**	0.8621	**0.4357**	–	1942
Prompts (phi3:mini, few shot, sample of 500)	0.3188	0.9167	**0.6764**	0.8718	**0.4311**	–	1901

Note. I consider macro F1 and PR–AUC the most meaningful metrics under class imbalance.

6 Reflection

Cost. The time needed to train and test the models was very different (Table 1). Embedding models trained in less than 15 seconds and were almost instant during testing. The transformer model took around 50 minutes to train and about 6 minutes to predict. Prompt-based models did not need training but were extremely slow at evaluation, taking around 30–35 minutes to process just 500 comments, which might simply depict the limitations of the used hardware. This means that simple linear or embedding models are best for quick use, while transformers and prompts are more suited when more resources are available.

Bias. Embedding models can sometimes mark neutral sentences like "She is a woman" as toxic. This happens because the embeddings are trained on large internet text that already contains social bias. As a result, these models can repeat or even amplify such bias if not checked or corrected.

Transparency. Logistic Regression is the easiest model to understand, since each word or feature has a clear weight showing how it affects the decision. Transformers are much harder to explain because their predictions come from many hidden layers. Prompt-based models are the least transparent since their reasoning happens inside a large language model that we cannot fully inspect.

Privacy. When using external APIs for prompting, the user comments have to be sent to third-party servers. This can create privacy risks, especially when working with user-generated text like Wikipedia comments. A safer option is to run the models locally or to anonymise the text before sending it.