Toxic Comment Classification

Luis Lloret Llinares
Date: August 2025

TABLE OF CONTENTS

1. EXPLORATORY DATA ANALYSIS

My initial approach focused on conducting a comprehensive Exploratory Data Analysis (EDA) to better understand the dataset before proceeding to model development. All experiments were tracked using **MLflow**, which I set up at the beginning of the notebook to ensure proper versioning, logging, and reproducibility of results. The code is located in the notebook 01_eda_multilabel.ipynb.

The main steps of the EDA included:

- **Descriptive statistics** of the dataset.
- **Label distribution analysis**, identifying how many samples correspond to each of the six target classes (toxic, severe_toxic, obscene, threat, insult, and identity_hate).
- **Label concurrence analysis**, which revealed how often multiple labels co-occurred in the same comment.
- **Correlation matrix** for the target labels to visualize interdependencies between different toxicity categories.
- **Distribution of label count per comment**, showing how many labels each comment typically has.
- **Comment length analysis by label**, examining the number of words in a comment depending on its label(s).

Key Findings:

- Most comments contain fewer than **50 words**, and the frequency of comments decreases as the word count increases.
- Interestingly, **toxic comments are more frequent among shorter texts**—particularly those between **0 and 25 words**—suggesting that toxic language tends to be **short and direct**.
- As the length of comments increases, the distinction between toxic and non-toxic comments becomes less pronounced. For comments with a higher word count, the frequency of toxic and non-toxic labels tends to converge.

- This insight implies that **word count could serve as a useful feature** for classification, especially to detect short and aggressive comments.
- Some labels, such as threat and identity_hate, are **rare and underrepresented**, making them difficult to model effectively due to their **low sample size** and less defined patterns.
- Overall, the dataset is **highly imbalanced**, which poses additional challenges for training robust classifiers.

2. BASELINE MODEL

In the second stage of the project, my objective was to **benchmark several pre-trained transformer models** as baselines for the Toxic Comment Classification task. This would allow me to later compare the impact of various preprocessing techniques and improvements. Code located at 02_baseline_hf_models.ipynb.

To do this, I created a selection of open-source transformer models available via Hugging Face, including:

- distilbert-base-uncased
- bert-base-uncased
- roberta-base
- microsoft/deberta-v3-base
- unitary/toxic-bert (a model specifically fine-tuned on the Jigsaw dataset)

Each model was evaluated **without any fine-tuning**, using a **balanced validation and test set** to ensure fair comparison across models.

Evaluation Metrics

For each model, I computed the following metrics:

- **Per-label AUC scores**
- **Mean AUC across all six toxicity categories**

- **Neutral sample accuracy**: percentage of clean (non-toxic) comments classified correctly
- **Neutral sample false positive rate**

Additionally, I applied **class weights** during evaluation to mitigate the effects of class imbalance in the dataset, especially for rare labels like threat and identity_hate.

Key Results

Model	Val AUC	Test AUC	Val Neutral Acc	Test Neutral Acc
unitary/toxic-bert	**0.9623**	**0.9377**	**1.0000**	**0.9050**
distilbert-base-uncased	0.5557	0.4589	0.0733	0.0000
microsoft/deberta-v3-base	0.5079	0.4896	0.0000	0.0000
roberta-base	0.5018	0.5379	0.0000	0.0000
bert-base-uncased	0.4861	0.4590	0.0000	0.0000

Observations:

- The unitary/toxic-bert model, which is fine-tuned specifically on the Jigsaw Toxic Comments dataset, **significantly outperformed** all general-purpose base models.
- All other models yielded **poor AUCs and extremely high false positive rates** on neutral comments, suggesting that pretraining alone is insufficient for this task, and **downstream fine-tuning is essential**.
- distilbert-base-uncased performed slightly better than other base models, likely due to its smaller architecture offering better generalization on limited data.
- The **neutral sample performance** was especially poor in all models except toxic-bert, confirming the difficulty of correctly identifying non-toxic content without task-specific fine-tuning.

These results establish unitary/toxic-bert as a **strong baseline** for future comparison. In the next steps, I will experiment with tagging, and fine-tunning to improve my performance.

3. TAGGING METHODS

This stage of the project aimed to evaluate whether the performance of the unitary/toxic-bert model, already fine-tuned on the Jigsaw Toxic Comment dataset,

could be further improved through the use of tagging strategies. The goal was to incorporate additional contextual or structural information directly into the text inputs and assess whether this would lead to better classification outcomes. Code located at 03_toxic_bert_tagging_methods.ipynb.

Methodology

Six distinct tagging strategies were implemented and evaluated:

1. **Baseline**
   No additional tagging was applied. This served as the control experiment.

2. **Explicit Markers**
   Direct toxicity labels (e.g., [THREAT], [INSULT]) were added to the beginning of comments based on known labels.

3. **Intensity Tagging**
   Comments were annotated with markers such as [INTENSITY:HIGH] or [CAPS:HEAVY] to signal potential aggression or emphasis.

4. **Target Tagging**
   Tags like [TARGET:PERSON] or [DIRECTION:DIRECT] were used to highlight who the comment was directed at and the nature of the attack.

5. **Linguistic Features**
   Structural attributes, such as [LENGTH:SHORT] or [QUESTION], were used to describe the comment's format.

6. **Semantic Context**
   Broader context tags (e.g., [CONTEXT:POLITICAL], [EMOTION:ANGER]) were introduced to reflect the topic or tone of the comment.

Each strategy was tested using a balanced dataset of 759 samples, with approximately 150 samples per class. Evaluation was based on per-class AUC, mean AUC, neutral comment accuracy, and false positive rate.

Results

Tagging Strategy	Mean AUC	Neutral Accuracy	Neutral False Positive Rate
Baseline	0.9562	99.33%	0.67%
All tagging variants	0.9562	99.33%	0.67%

Per-label AUC (Baseline):

- toxic: 0.9811
- severe_toxic: 0.8824
- obscene: 0.9813
- threat: 0.9687
- insult: 0.9536
- identity_hate: 0.9701

Key Findings

- None of the tagging strategies resulted in any improvement over the baseline. All configurations produced identical results.
- The performance of the unitary/toxic-bert model is already very high, suggesting it is well optimized for the dataset.
- Possible explanations for the lack of improvement include:
    - The model has already learned the semantic cues that the tags attempt to introduce.
    - Adding tags may introduce noise rather than clarity, particularly if the tokenizer is not designed to handle custom tokens.

- The experiment was conducted on a balanced dataset to mitigate label imbalance. However, minority classes like threat (only 50 samples) remain a challenge for any modeling strategy.

Conclusions

- The baseline model (unitary/toxic-bert) performs exceptionally well without any need for additional tagging.
- Tagging strategies, although theoretically informative, did not contribute to improved performance in this context.
- The model achieved a high mean AUC (95.62%) and a very low false positive rate on neutral comments (0.67%).

4. FINE-TUNNING

This fase focused on evaluating whether fine-tuning the unitary/toxic-bert model on a balanced subset of the dataset could yield measurable improvements in classification performance. Additionally, it explored whether incorporating subtle tagging techniques during training would enhance the model's sensitivity to more nuanced forms of toxicity.

Three experimental configurations were compared:

1. **Baseline**: Pretrained unitary/toxic-bert model with no additional training.
2. **Fine-Tuned**: Toxic-BERT fine-tuned on a balanced training dataset.
3. **Fine-Tuned + Tagged**: Toxic-BERT fine-tuned on the same dataset, augmented with subtle toxicity tagging (SUBTLE_TOXICITY method).

Methodology

1. **Data Preparation**
    - A balanced subset of the dataset was used: approximately 250 samples per label, resulting in around 1,680 total samples.

- Dataset splits:
  - Train: 1,176 samples
  - Validation: 168 samples
  - Test: 336 samples
- Training configuration:
  - max_length = 128
  - batch_size = 64
  - epochs = 1 (to reduce training time)

2. **Tagging Method: SUBTLE_TOXICITY**
   This strategy aimed to tag phrases with indicators of subtle toxic behavior. Patterns included:
   - **Passive-aggressive**: e.g., *"just saying", "whatever"*
   - **Sarcastic**: e.g., *"oh really", "how original"*
   - **Disparaging**: e.g., *"pointless", "waste of time"*
   - **Condescending**: e.g., *"let me explain", "actually"*

Tagging coverage:

- 71 out of 1,176 training samples were tagged (6.04%)
- Medium intensity: 67 samples (5.70%)
- High intensity: 4 samples (0.34%)

Performance Comparison

Model Configuration	**Mean AUC**	**Neutral Accuracy**	**Neutral False Positive Rate**
Baseline	0.9526	1.0000	0.0000

Model Configuration	Mean AUC	Neutral Accuracy	Neutral False Positive Rate
Fine-Tuned	0.9540	0.9808	0.0192
Fine-Tuned + Tagged	0.9525	0.9808	0.0192

AUC by Class

Model	toxic	severe_toxic	obscene	threat	insult	identity_hate
Baseline	0.9950	0.8364	0.9759	0.9823	0.9458	0.9801
Fine-Tuned	0.9948	0.8433	0.9761	0.9828	0.9470	0.9802
Fine-Tuned + Tagged	0.9955	0.8409	0.9750	0.9788	0.9455	0.9795

Key Findings

1. **Marginal Gains from Fine-Tuning**
    - Fine-tuning led to a **+0.15% increase** in mean AUC compared to the baseline.
    - The most notable improvements were in the severe_toxic (+0.69%) and threat (+0.05%) categories.
    - However, this came at the cost of reduced precision in neutral comment classification, with a **1.92% false positive rate**.
2. **Limited Impact of Subtle Tagging**
    - The model trained with SUBTLE_TOXICITY tagging showed **no improvement** over the baseline or standard fine-tuning.

- Compared to the baseline, performance slightly decreased by **-0.01%** in mean AUC.
- Compared to the fine-tuned model, the drop was **-0.16%**, suggesting that tagging introduced noise rather than meaningful signal.

3. **Strong Baseline Performance**
   - The pretrained unitary/toxic-bert already demonstrated very high performance (Mean AUC: 95.26%).
   - These results further confirm that it is difficult to improve an already highly optimized model without significant dataset or architecture changes.

5. POTENTIAL FUTURE IMPROVEMENTS

If provided with a longer timeframe, several enhancements could be explored to increase model performance:

1. **Extensive Fine-Tuning on a larger dataset**
   While the current fine-tuning was conducted on a small, balanced subset for efficiency, scaling this to larger and more representative dataset could improve the model's generalization, particularly for underrepresented toxic categories such as *threat* or *identity_hate*.

2. **Advanced Data Augmentation Techniques**
   Techniques like synonym replacement, or paraphrasing could be used to synthetically expand the dataset, especially for minority classes. This would mitigate the effects of class imbalance and improve model sensitivity to rare but critical toxicity types.

3. **LLMs**
   An alternative line of research could explore prompt-based toxicity detection using models like GPT. This might allow better handling of context-dependent toxicity and leveraging on models trained on huge datasets.

4. **Improved Tagging Methodologies**
   While tagging strategies like SUBTLE_TOXICITY showed limited impact, more sophisticated tagging approaches—potentially informed by linguistic and psychological research on hate speech—could improve detection of implicit or veiled toxicity.

6. CHALLENGES IN DEPLOYMENT

Deploying a toxicity detection system in a user-facing environment involves more than achieving high performance metrics in a controlled experimental setting. Several key challenges must be addressed to ensure the system is both effective and ethically sound in real-world use.

1. Balancing Precision and Recall: False Positives vs. False Negatives

Even with high AUC scores, small error margins can have significant consequences. False positives—where benign content is flagged as toxic—may frustrate users, raise concerns about censorship, and undermine trust in the platform. Conversely, false negatives—where toxic comments go undetected—pose serious risks, particularly when dealing with harmful categories such as threats, hate speech, or harassment. This balance depends on the objective of the solution.

2. Generalization to Real-World and Evolving Language

Models trained on well-curated and balanced datasets may struggle to generalize to the diversity and messiness of real-world content. Language on public platforms evolves rapidly, with new slang, coded terms, and multilingual expressions appearing regularly. Without frequent retraining or adaptive learning strategies, the system risks becoming outdated.

3. Operational Constraints and Scalability

In production environments with high user activity, the system must be capable of processing large volumes of content in real time without introducing noticeable delays. Any lag in moderation workflows can negatively impact user experience and platform responsiveness. To ensure efficient and scalable inference, performance optimization

becomes essential. This may involve leveraging lighter model architectures through distillation, or deploying on high-performance computing resources such as GPUs.