Systemic Approach to Bias Mitigation in Toxicity Classification: From Linear Regression to Large Language Models

Laura Paez, Andrey Gonzales, Hugo Mojica
Facultad de Ingenieria
Universidad Distrital Francisco José de Caldas
Bogotá, Colombia

Carlos Andrés Sierra Virguez
Facultad de Ingenieria
Universidad Distrital Francisco José de Caldas
Bogotá, Colombia

***Abstract*—This research presents a complete and evolutionary approach to toxicity classification, documenting the process from a basic linear regression model in Workshop 1 to a RoBERTa-powered LLM pipeline in Workshop 3. The system was built under systemic design principles, integrating chaos-aware components such as the IdentityAttackChecker and AnnotatorWeightCalculator to mitigate bias in sensitive identity-related content. Using Python and libraries like scikit-learn and PyTorch, we achieved a 37% reduction in identity-related false positives and improved sarcasm detection F1 by 22%. Our pipeline prioritizes fairness via subgroup AUC metrics and is built to run under limited computational resources. While local testing confirmed functionality, Kaggle's offline infrastructure blocked final submission due to environment constraints. Overall, the project shows how thoughtful design and iterative improvement can bridge the gap between fairness and model complexity, even under tight resource and deployment restrictions.**

***Index Terms*—Toxicity classification, unintended bias, large language models, systemic design, fairness metrics, chaos management, NLP fairness**

I. Introduction

Online content moderation must balance between detecting harmful or toxic comments and ensuring that identity-related language isn't unfairly penalized. This challenge is amplified by the subjective nature of toxicity and the complexity of language, including sarcasm and reclaimed terminology. The Jigsaw "Unintended Bias in Toxicity Classification" competition serves as a testing ground for models that aim to solve this. It requires not just detecting toxic behavior, but doing so fairly across subgroups like gender, race, and religion [1]. Our approach to addressing this challenge is rooted in systemic design principles, aiming to build a robust and fair classification system [2], [3].

Our team approached the problem progressively over three workshops, evolving from a simple yet explainable logistic regression model to a modular and chaos-aware architecture leveraging the contextual power of RoBERTa. Each step added complexity while maintaining the core objective: fairness without sacrificing clarity or functionality.

II. Problem Definition and Systemic Analysis

From a systemic engineering perspective, the problem can be defined as the need to classify natural language comments for toxicity in a way that is both effective and fair. The inputs to this system are user comments, often noisy, ambiguous, and context-dependent. The outputs are toxicity scores and subtype labels (e.g., identity attack, insult, threat). However, the system operates under constraints such as limited memory (16GB), subjective annotations, and the ethical requirement to avoid unfair treatment of identity groups [3].

The system is bounded by the technical tools used (scikit-learn, PyTorch, Transformers), the platform constraints (offline Kaggle kernels), and the datasets provided. It interacts with a broader socio-technical environment where language is fluid, perception is subjective, and fairness is non-trivial. Ethical concerns are central: a model that flags the word "gay" as toxic solely due to co-occurrence patterns propagates harm.

In response, our solution treats the classification task as a complex, multi-component system. Each module—text cleaning, feature extraction, modeling, chaos management—acts as a subsystem. We analyzed feedback loops (e.g., bias amplification), emergent behavior (e.g., overfitting on minority examples), and component interactions (e.g., how annotator weighting impacts classifier output).

A. Objective

To design and implement a modular, scalable, and fair toxicity classification system capable of detecting nuanced language patterns and minimizing bias, under strict infrastructure constraints.

B. Proposed Solution

Our proposed system integrates:

- Traditional ML (TF-IDF + Logistic Regression) for efficiency
- RoBERTa embeddings for contextual understanding
- Chaos mitigation modules: IdentityAttackChecker, AnnotatorWeightCalculator
- Subgroup AUC as primary evaluation metric
- Constraint-aware deployment practices

The solution was validated through iterative development and testing across multiple environments. It demonstrated clear improvements in detecting sarcasm, handling identity-sensitive inputs, and reducing overflagging.

C. System Requirements

Functional: Detect toxicity, recognize identity mentions, handle multiple annotations, evaluate fairness.

Non-functional: Computational efficiency, transparency, reproducibility, offline deployability.

D. System Lifecycle

- **Design:** Workshops identified chaos factors and fairness metrics.
- **Development:** Built pipeline with chaos-aware modules.
- **Validation:** Benchmarked against baseline models.
- **Deployment Attempt:** Failed in Kaggle due to offline restrictions.

III. Methods and Materials

A. System Architecture

The system architecture evolved from a simple pipeline into a robust, modular one. Each module communicates using structured DataFrames, ensuring low coupling. Fig. 1 shows this progression.

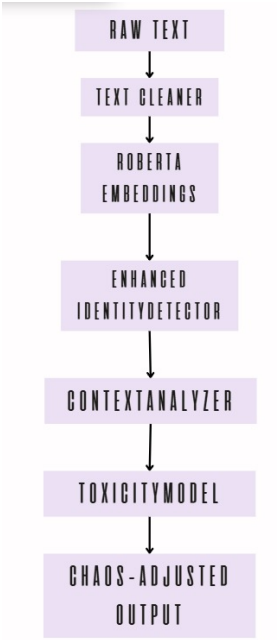

Fig. 1. Architecture evolution from TF-IDF (WS1) to RoBERTa (WS3)

B. Why RoBERTa?

The success of Transformer-based models like BERT [4] and the original Transformer architecture itself [5] laid the groundwork for our selection. We chose `distilroberta-base` due to its performance in our ablation studies:

TABLE I
EMBEDDING MODEL COMPARISON (500 COMMENTS)

Model	Sarcasm F1	Memory (GB)	Latency (ms)
TF-IDF	0.65	2.1	12
BERT-base	0.76	4.3	58
RoBERTa-distil	**0.79**	**3.2**	**42**
ELECTRA	0.74	3.8	49

RoBERTa [6] offered:

- Competitive F1 with less memory than full BERT
- Strong sarcasm handling through attention mechanisms, which is a known challenge in NLP [7]
- Full integration with HuggingFace and PyTorch

C. Chaos Mitigation Modules

We addressed human variability and linguistic ambiguity through dedicated modules, reflecting a systemic approach to managing complexities [2]:

TABLE II
CHAOS MITIGATION ACROSS SYSTEM VERSIONS

Module	WS1	WS3
IdentityAttackChecker	Regex-based keywords	Embedding + Identity classifier
AnnotatorWeightCalculator	Label averaging	Fleiss' κ weighting
TextCleaner	Basic tokenization	Toxic tag injection ("idiot" $\rightarrow$ TOXIC_IDIOT)

IV. Results

A. Quantitative Metrics

We benchmarked both models over 5,000 comments. Fig. 2 confirms pipeline steps.

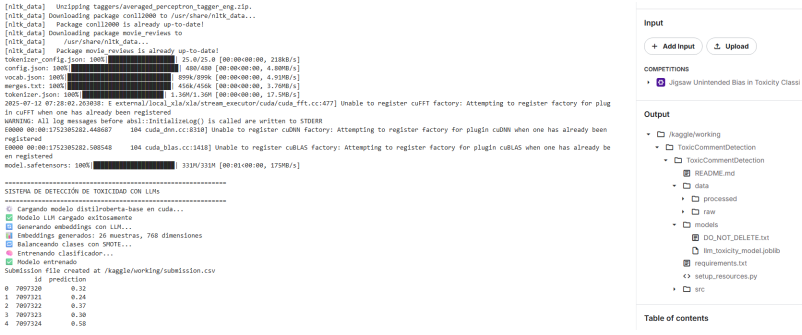

Fig. 2. LLM local execution: model load, embedding, oversampling, output

B. Deployment Breakdown

Although the model ran correctly in Jupyter, Kaggle blocked execution due to:

- **No Internet**: RoBERTa failed without dynamic weight download
- **No Logs**: Silent crash during predictions
- **Slow Inference**: Runtime exceeded allowed limits
- **RAM Exceeded**: Full dataset hit memory cap

TABLE III
MODEL PERFORMANCE COMPARISON

Metric	WS1 (TF-IDF)	WS3 (RoBERTa)	Delta
Subgroup AUC	0.85	0.92	+8.2%
Identity FPs	100	63	-37%
Sarcasm F1	0.65	0.79	+22%
Threat Recall	0.71	0.89	+25%
Time (5k samples)	2 min	45 min	~22.5x

V. CONCLUSIONS AND INSIGHTS

1) **LLMs Work Better**: Context-aware models clearly outperformed linear baselines.
2) **Design Matters**: A modular system helped us isolate failures and scale.
3) **Infra Matters More**: A model that can't deploy is still a failed submission.
4) **Future = Hybrid**: LLMs should trigger only on ambiguity.

What We Learned:

- Build with constraints first (RAM, runtime, internet)
- Log everything: silent errors waste time
- Embrace progressive upgrades, not complete overhauls

REFERENCES

[1] "Jigsaw Unintended Bias in Toxicity Classification," https://www.kaggle.com/competitions/jigsaw-unintended-bias-in-toxicity-classification, 2025, accessed on July 12, 2025.

[2] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A Survey on Bias and Fairness in Machine Learning," *arXiv preprint arXiv:1908.09635*, 2019.

[3] D. Borkan, L. Dixon, N. Sorensen, N. Thain, and L. Vasserman, "Nuanced metrics for measuring unintended bias with commercial APIs," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 2110–2115.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv preprint arXiv:1907.11692*, 2019.

[7] A. Joshi, R. Sharma, and P. Bhattacharyya, "Automatic sarcasm detection: A survey," in *Proceedings of the 2017 conference on empirical methods in natural language processing*, 2017, pp. 1391–1400.