FNLP412@EEUCA 2026: Understanding Toxic Behavioral Intent in Gaming Chat Logs using Transfer Learning and Synthetic Data Augmentation

Mihai Radu Rădulescu
University of Bucharest
mihai-radu.radulescu@s.unibuc.ro

Abstract

Our paper explores several machine learning methods for detecting toxic language in gaming-related chat utterances. We start with the GameTox dataset, perform some data preprocessing and augment the minority classes with LLM-generated synthetic data. We then set a baseline using a classic Logistic Regression model and continue to explore several approaches to surpassing it, by leveraging the leading multilingual transformer models (XLM-RoBERTa and DeBERTa-V3) to classify our test data. We achieve a top result of 0.6725 Macro-F1 (2nd place on shared task leaderboard) using a MDeBERTa-V3 model which we pretrained on the Jigsaw dataset for 1 epoch and then fine-tuned on our GameTox data for 5 epochs.

1 Introduction

Toxic behavior and messaging in online multiplayer games is widespread. This paper explores designing and training several machine learning models to help implement robust detection methods to ensure user safety. We have written this as part of a shared task (Thapa et al., 2026) within the EEUCA workshop (Hürriyetoğlu et al., 2026) (ACL 2026). Our results show that, of several approaches considered, the best performing one is a DeBERTa-V3 model which we pretrained on a more generic toxicity dataset (Kivlichan et al., 2020), achieving a Macro-F1 score of 0.6725 (2nd place on the final leaderboard).

2 Related Work

Since the EEUCA shared task is still ongoing, there is no established SOTA or baseline regarding the GameTox dataset. In the original GameTox paper (Naseem et al., 2025), the authors identify Joint BERT (Chen et al., 2019) as the most promising model, with a 0.89 accuracy score. On a similar gaming chat dataset, CONDA (Weld et al., 2021), the winning paper (Jia et al., 2024) describes the BRAR model, which implements attention residuals, slot filling and label forcing. ToxBuster (Yang et al., 2023) is a model which reaches an accuracy of 0.82 by creating context from groups of sequential chat messages (not possible for our dataset). ToXCL (Hoang et al., 2024) is a framework for toxicity detection using Contrastive Loss to address dataset class imbalance issues with good results. However, its applicability to gaming chat is not as performant.

3 Method

3.1 Dataset

The shared task is based on GameTox (Naseem et al., 2025), a dataset of chat utterances collected from the game "World of Tanks". The dataset contains messages labeled on a scale from 0 to 5 (an annotation schema shared with Crisishatemm (Bhandari et al., 2023)), with the labels indicating the following mapping:

- 0 - Non-toxic[4]
- 1 - Insults and flaming[5]
- 2 - Other offensive texts[6]
- 3 - Hate and harassment[7]
- 4 - Threats[8]
- 5 - Extremism[9]

The messages are generally very short, with an average length of 13.84 characters and a 75th percentile of 18 characters (Figures 2 and 3). This brevity presents additional challenges for effective training and detection. Another noteworthy characteristic is the multilingual aspect of the dataset: messages are mostly in English, but also in Russian, German, Polish, French, and other languages, making it essential to use models capable of handling multiple languages.

Proceedings of the 9th Workshop on Event Extraction and Understanding: Challenges and Applications, pages 96–103

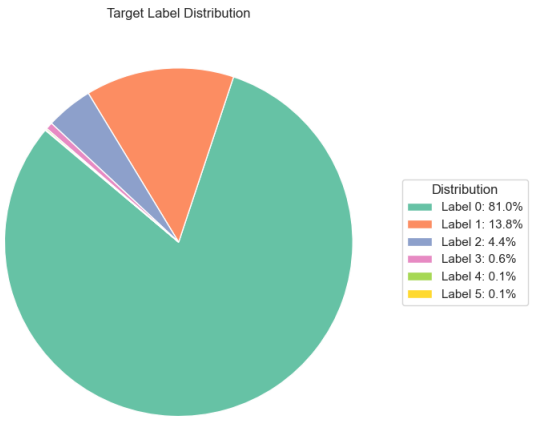

Figure 1: Label distribution

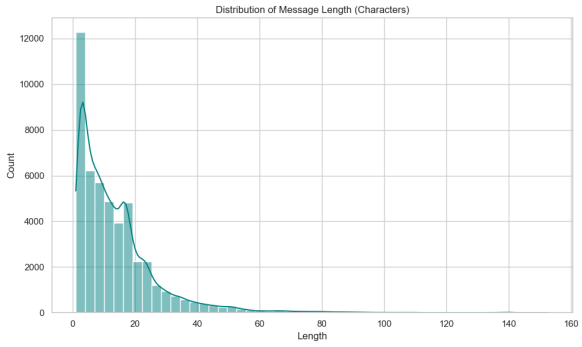

Figure 2: Message length

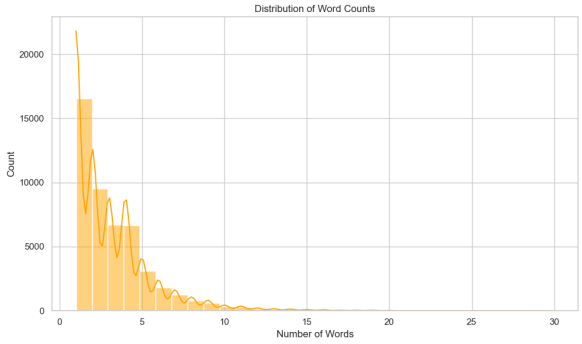

Figure 3: Word count

The dataset is highly imbalanced, with the majority of messages being labeled as non-toxic (81%), down to 0.06% labeled 5 - only 27 samples - Figure 1. To address this imbalance, we employed several preprocessing techniques to clean and normalize the text data, as detailed below.

3.2 Preprocessing

After exploring the dataset, we put together the following steps for preprocessing the message data:

1. **URL removal**: we replaced all URLs with a [URL] tag, to avoid URL content influencing our training.
2. **Lowercase**: we converted all text to lowercase to ensure consistency.
3. **User mentions**: we replaced all @user mentions/tagging with a common [USER] tag.
4. **Repetition normalization**: we reduced characters repeated more than twice. This keeps the emphasis, but removes redundancy. (e.g. 'looooool' -> 'lool')
5. **Slang map**: we applied a mapping for common slang, abbreviations, typos and obfuscations, most of which we manually extracted from the messages in classes 3-5.
6. **Augmentation**: we generated 50 synthetic data samples for classes 4-5, where our original dataset had very few examples. We used Gemini 3 Pro with the prompt in Appendix A to generate these samples.

3.3 Models

3.3.1 M1: Basic baseline

Since the shared task does not provide a baseline, we implemented a simple Logistic Regression model using TF-IDF (Spärck Jones, 1972) features. This model serves as a reference point for evaluating the performance of more complex models in our next attempts.

Our baseline Macro-F1 score, using LR, was 0.5046 on the validation set and 0.4665 on the test set.

3.3.2 M2: XLMR (0.3B parameters)

As a next step, we fine-tuned the XLM-RoBERTa-base transformer model (Conneau et al., 2019) on the GameTox dataset. We chose XLMR because it is a multilingual model pre-trained on 100 languages, making it suitable for our task which includes messages in multiple languages. We used the HuggingFace Transformers library for implementation, and trained the model for 5 epochs with

a learning rate of 2e-5 and a batch size of 16. Prior to training, we performed a 5-fold stratified split of the training data (ensuring class distribution is maintained in each fold), and used 4 folds as training data and 1 fold as validation data, over 5 iterations. We then averaged the results across the folds to obtain a robust estimate of model performance.
Our fine-tuned XLMR model achieved a Macro-F1 score of 0.5297 on the validation set, and 0.5752 on the test set. At this point, even while this was the leading model in the shared task leaderboard, we aimed to push performance even higher.

3.3.3 M3: XLMR pretrained on Jigsaw

To improve our score, we next adopted a transfer learning approach. We first pre-trained the XLM-RoBERTa-base model on the Jigsaw Multilingual Toxic Comment Classification dataset (Kivlichan et al., 2020), which contains a large number of toxic comments in multiple languages. This pre-training step helps the model learn general features of toxic language, which can then be fine-tuned on the more specific GameTox dataset. After pre-training on Jigsaw for 1 epoch, we fine-tuned the model on GameTox for 5 epochs using the same hyperparameters as before. Since Jigsaw is a much larger dataset, its pre-training should improve our model's sensitivity to detect toxicity in written form. However, Jigsaw is not specifically focused on gaming chat, so we transferred this knowledge to the specifics of our GameTox dataset in the subsequent training step.
The result was a Macro-F1 score of 0.4755 on the test set, which was lower than our previous XLMR model. This indicates that while transfer learning can be beneficial, it may not always lead to improved performance, especially if the pre-training dataset is not closely aligned with the target task.

3.3.4 M4: MDeBERTA-V3 (0.27B parameters) pretrained on Jigsaw

Since the first transfer learning attempt attained lower performance than expected, our next approach involved switching to a different transformer architecture: Microsoft DeBERTa-V3 (He et al., 2021), which has shown strong performance on various NLP tasks, and is still multilingual. We followed the same transfer learning approach as before, pre-training the DeBERTa-V3-base model on the Jigsaw dataset for 1 epoch, and then fine-tuning it on GameTox for 5 epochs. We used a learning rate of 2e-5 and a batch size of 16. Specifically for the pretraining step, we increased the maximum input length to 256 tokens (from the default 64), since many Jigsaw comments are longer and would be otherwise truncated. This allows the model to capture more context from longer comments, which could be beneficial for toxicity detection.
Our model achieved a Macro-F1 score of 0.6258 on the validation set, and 0.6725 on the test set. It is our best-scoring model so far, and remained our top contender, but we tried a few other approaches to see if we can improve on our performance.

3.3.5 M5: Ensemble prediction

Since our models are showing different strengths and weaknesses (higher precision vs higher recall), we decided to combine their predictions using an ensemble approach. We used a weighted average of the predicted probabilities of our **M3** and **M4** models, with weights determined based on their validation performance (F1 score).
This resulted in a Macro-F1 of 0.6165 - not bad, but lower than the simple predictions of **M4**.

3.3.6 M6: Pseudo label augmentation of M4

Our last improvement idea involved another type of data augmentation, where we used our leading model as a "teacher" model to train a fresh "student" model. We again used the DeBERTa-V3-base model as the student, and trained it to mimic the predictions of the **M4** model on the training data. This involved merging the original training data (triplicated, to keep it dominant) with the pseudo-labeled data generated by the teacher model.
We then trained the student model on this merged dataset for 3 epochs, obtaining a Macro-F1 score of 0.5927 on the test set, and 0.6114 after threshold optimization (see 3.3.8).

3.3.7 M7: Pseudo label augmentation of M5

The final model we experimented with was trained similarly to **M6**, but on the predictions generated by **M5** instead of **M4**. The result was a Macro-F1 score of 0.6311, still below our **M4** model.

3.3.8 Threshold Optimization

At inference time, by default, calculating a prediction would mean picking the class label with the highest predicted probability. Given the imbalanced nature of our dataset, we decided to optimize the decision thresholds for each class to improve recall on the less represented classes. We performed a grid search over [0.10, 0.95] threshold values

for each class and determined the optimal thresholds to maximize the F1 score. Next, during the prediction generation, we implemented a severity waterfall, and checked each predicted probability against each class threshold, in descending order (5 to 0). If the probability exceeded a threshold, we chose this class as our final prediction and skipped to the next sample.
Some models reacted favorably to the threshold tuning, while others did not. Where not explicitly specified, we chose the results with the higher Macro-F1 score of the two.

4 Future Work

Strong class imbalance, such as found within our dataset, has been shown to be well handled by Focal Loss (Lin et al., 2018) and Adversarial Weight Perturbation (Wu et al., 2020) techniques.
Another avenue worth exploring is using Joint Intent Classification and Slot Filling (Chen et al., 2019) - we are ignoring slots in our implementation at this point and haven't explored this further.
On the data augmentation front, there are several approaches which could yield interesting results. One is to go beyond synthetic data generation using LLMs, and explore backtranslation (Sugiyama and Yoshinaga, 2019) or other synthetic generation methods.
Similar to how we used the Jigsaw dataset, one could search for another dataset to use for pretraining the transformers during the first epoch - perhaps the CONDA dataset (Weld et al., 2021) or other toxicity-focused datasets.
Model distillation between architectures (e.g. using XLMR as a teacher and DeBERTa as student, or using a monolingual model as the student for a multilingual teacher) was also not explored due to time constraints, and might be a promising avenue.
Finally, there are other multilingual transformer architectures, such as BERT or other sizes of XLMR/MDeBERTa, which could yield different results.

5 Conclusion

Of all the various model and parameter combinations we experimented with, our best result was obtained using model **M4** (3.3.4) - a Macro-F1 score of 0.6725 in the shared task competition, which placed us at position #2 in the leaderboard at the end of the test phase.
A comparative table detailing the performance obtained with the various implemented models can be seen in Table 1.

Our present research on the topic of toxicity in online gaming chat shows that DeBERTa-V3 is a capable multilingual model for this task, and pretraining it on a toxicity-related dataset (even if not gaming focused) surpasses other implementations using straightforward training or other transformer models.

Limitations

The transformers we have used for this task, such as XLM-RoBERTa and DeBERTa-V3, are relatively large models (approx. 0.3B parameters) that require significant computational resources for both training and inference. On a MacBook M4 Pro, training an epoch can take one hour, and the chances for improving hyperparameters or experimenting with model modifications are limited by time. Additionally, while these models are pretrained on multiple languages, their performance may vary significantly across different languages present in the dataset - this is a limitation we did not explore in this project. When we pretrained the XLMR model (M3 3.3.3), we used the same max length of 64 as for the GameTox dataset. However, Jigsaw text is much more ample, thus a max length of 256 would have been more appropriate, ensuring less content is truncated. Larger models (such as DeBERTa-V3-large or XLM-RoBERTa-XL) could potentially yield better results, but their training would not have been feasible within our resource constraints.

Ethical Considerations

The datasets processed in this research contain highly offensive, hateful, and disturbing language, including threats, hate speech, severe profanity and extremism. Our models are discriminative (classifiers) rather than generative; they do not generate new toxic text, but serve only to flag existing toxicity. While this technology is designed to foster safer online communities, we recognize the dual-use potential of automated moderation tools. If misapplied, such systems could be used for censorship or surveillance. We emphasize that our models are intended to assist human moderators by flagging potential violations, not to act as autonomous decision makers without human oversight.

	M1	M2	M3	M4	M5	M6	M7
F1 Macro	0.4665	0.5752	0.4755	0.6725	0.6165	0.6311	0.6114
Accuracy	0.8804	0.8739	0.8448	0.8992	0.8964	0.8794	0.8954
Precision	0.4561	0.526	0.4162	0.6636	0.6135	0.5783	0.5695
Recall	0.5029	0.6521	0.6588	0.6846	0.6237	0.7194	0.6844

Table 1: Training results

References

Aashish Bhandari, Siddhant B Shah, Surendrabikram Thapa, Usman Naseem, and Mehwish Nasim. 2023. Crisishatemm: Multimodal analysis of directed and undirected hate speech in text-embedded images from russia-ukraine conflict. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1994–2003.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *Preprint*, arXiv:1902.10909.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Nhat M. Hoang, Xuan Long Do, Duc Anh Do, Duc Anh Vu, and Luu Anh Tuan. 2024. Toxcl: A unified framework for toxic speech detection and explanation. *Preprint*, arXiv:2403.16685.

Ali Hürriyetoğlu, Surendrabikram Thapa, Hristo Tanev, Laxmi Thapa, and Surabhi Adhikari. 2026. Overview of the workshop on event extraction and understanding: Challenges and applications. In *Proceedings of the 9th Workshop on Event Extraction and Understanding: Challenges and Applications (EEUCA)*.

Yuanzhe Jia, Weixuan Wu, Feiqi Cao, and Soyeon Caren Han. 2024. In-game toxic language detection: Shared task and attention residuals (student abstract). *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(13):16238–16239.

Ian Kivlichan, Jeffrey Sorensen, Julia Elliott, Lucy Vasserman, Martin Görner, and Phil Culliton. 2020. Jigsaw multilingual toxic comment classification. `https://kaggle.com/competitions/jigsaw-multilingual-toxic-comment-classification`. Kaggle.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2018. Focal loss for dense object detection. *Preprint*, arXiv:1708.02002.

Usman Naseem, Shuvam Shiwakoti, Siddhant Bikram Shah, Surendrabikram Thapa, and Qi Zhang. 2025. Gametox: A comprehensive dataset and analysis for enhanced toxicity detection in online gaming communities. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 440–447.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

Amane Sugiyama and Naoki Yoshinaga. 2019. Data augmentation using back-translation for context-aware neural machine translation. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 35–44, Hong Kong, China. Association for Computational Linguistics.

Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Kritesh Rauniyar, Laxmi Thapa, Surabhi Adhikari, Kristina T. Johnson, Ali Hürriyetoğlu, Hristo Tanev, and Usman Naseem. 2026. Understanding toxic behavior in gaming communities using ai to promote healthier digital spaces. In *Proceedings of the 9th Workshop on Event Extraction and Understanding: Challenges and Applications (EEUCA)*.

Henry Weld, Guanghao Huang, Jean Lee, Tongshu Zhang, Kunze Wang, Xinghong Guo, Siqu Long, Josiah Poon, and Soyeon Caren Han. 2021. Conda: a contextual dual-annotated dataset for in-game toxicity understanding and detection. *Preprint*, arXiv:2106.06213.

Dongxian Wu, Shu tao Xia, and Yisen Wang. 2020. Adversarial weight perturbation helps robust generalization. *Preprint*, arXiv:2004.05884.

Zachary Yang, Nicolas Grenon-Godbout, and Reihaneh Rabbany. 2023. Towards detecting contextual real-time toxicity for in-game chat. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics.

A Synthetic data prompt

Prompt used for synthetic data generation (using Gemini 3 Pro):

"I am training an NLP model to detect 'Extremism' and 'Threats' in online gaming chat (World of Tanks).

Please generate:

50 distinct examples of 'Extremism' (Class 5): These should focus on radical ideology, promotion of terrorist groups, or extreme political hate, but written in 'gaming chat style' (short, lowercase, maybe some typos).

50 distinct examples of 'Threats' (Class 4): These should be physical threats of violence (e.g., 'I will find where you live', 'kill yourself'), distinct from simple insults.

Format the output as a CSV with columns: message,label (where label is 5 for Extremism and 4 for Threats)."

B Word clouds of each label class

Figure 4: 0 - Non-toxic

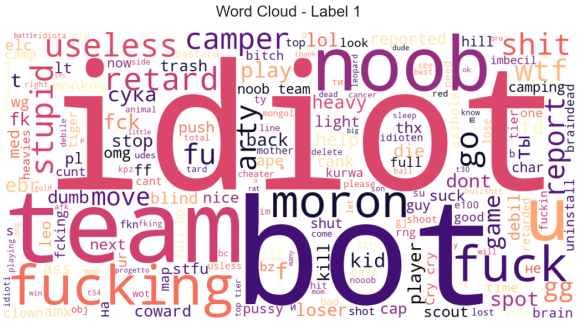

Figure 5: 1 - Insults and flaming

Figure 6: 2 - Other offensive texts

Figure 7: 3 - Hate and harassment

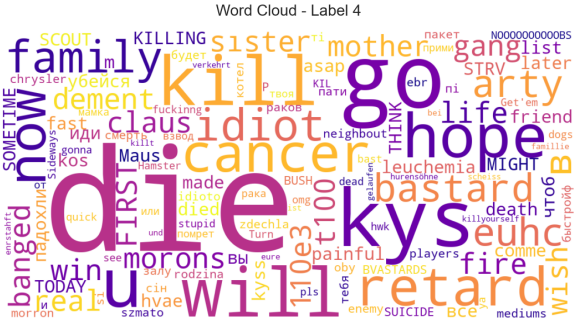

Figure 8: 4 - Threats

Figure 9: 5 - Extremism