TAGA@EEUCA 2026: Token-Attribution Guided Attention for Fine-Grained Toxic Behaviour Classification in Online Gaming Communities

Akshyat Shah and Shashi Sah and Aryan Gupta and Kavinder Singh
Delhi Technological University
Delhi, India
{akshyatshah, sah.shashi2003, aryangupta0419, kavinder85}@gmail.com

Abstract

Online gaming involves large amount of people forming a large community of players who interact in real time. Toxic behavior in online chat is common and can harm players by deterring them. Thus, automated moderation is a necessity but difficult because game chat mixes domain-specific slang, deliberate obfuscation, informal "gamer" language , and limited coverage for categories such as threats and extremism. This paper describes the TAGA (Token-Attribution Guided Attention) system submitted to the EEUCA 2026 Shared Task on Understanding Toxic Behavior in Gaming Communities. We propose TAGA, an architecture that employs a leave-one-out attribution method using the Detoxify toxicity scorer to compute per-token attribution scores across multiple toxicity dimensions, which are then projected into the learned attention biases that steer the model toward toxicity-indicative tokens. By preparing a five phase ablation study, we demonstrate that each component: domain-specific preprocessing, focal loss with label smoothing, attribution-guided attention pooling, and dual-model Detoxify features with strategic oversampling contributes to a cumulative gain in macro-F1 score points over the DeBERTa-v3-base baseline reported. The final system achieves a test macro-F1 score of 0.618 and, importantly, produces non-zero predictions for extreme data imbalance present in the dataset used in the shared task.

1 Introduction

The rapid rise of the culture of online gaming has created several communities where millions of players interact with each other in real time over chat. While most of these interactions are fruitful and often positive, toxic behavior such as harassment, threats, hate speech, and extremism, still remains prevalent and can cause significant mental harm, psychological harm and damage to players (Kwak et al., 2015; da Silva et al., 2020). Hate speech detection using NLP has seen significant improvements with the advancements in language models (Parihar et al., 2021), yet moderating game chat at scale remains challenging due to the unique linguistic characteristics and gamer speak of the community.

Gaming chat exhibits several linguistic properties that distinguish it widely from standard social media text: extensive use of game-specific slang, deliberately obscuring chats through leetspeak substitutions (e.g., “naz1” for “nazi”), and highly compressed messages where a single word may carry the entire toxic intent (Märtens et al., 2015; Blackburn and Kwak, 2014). Furthermore, the distribution of toxicity categories is severely imbalanced: in the GameTox dataset (Naseem et al., 2025), non-toxic messages constitute over 81% of the data, while critical categories like Threats (0.14%) and Extremism (0.06%) contain fewer than 60 samples each. The annotation schema follows the hate speech categorization framework established in Bhandari et al. (2023).

The EEUCA 2026 Shared Task on Understanding Toxic Behavior in Gaming Communities (Thapa et al., 2026), which is organized as a part of the 9th Workshop on Event Extraction and Understanding (Hürriyetoğlu et al., 2026), provides a platform for developing and benchmarking automated toxicity detection systems on this challenging set of data. Naseem et al. (2025) introduced GameTox, which is a dataset consisting of 53K game chat utterances from the World of Tanks game and annotated for both utterance-level intent classification (6 classes) and token-level slot filling (4 slot types). Their baseline experiments showed that using Joint BERT (Chen et al., 2019) achieved the best performance among the 12 baseline models which were evaluated. However, the gap between the performance of slot-filling (0.99 F1) and intent classification (0.89 F1) suggests that understanding the overall intent of a message, particularly for rare

Proceedings of the 9th Workshop on Event Extraction and Understanding: Challenges and Applications, pages 198–207

categories, still remains as a challenging task.

In this work, we propose **TAGA**, our submission to the EEUCA 2026 shared task. TAGA combines token-level toxicity signals and utterance-level intent classification through an attribution-guided attention mechanism. Rather than relying on manually annotated slot labels, TAGA automates computation of token-level toxicity attributions using leave-one-out token approach with the Detoxify toxicity scorer (Hanu and Unitary team, 2020). These attributions are then projected into the attention bias terms that guide the model toward determining the tokens which are the most indicative of the utterance's toxicity. Our key contributions are:

1. A **token-attribution guided attention** mechanism that injects externally computed toxicity signals into the attention bias and computes using a pre-trained language model.

2. A **multi-channel attribution** approach using leave-one-out perturbation across four toxicity classes to capture fine-grained token-level signals.

3. A comprehensive **domain-specific preprocessing** pipeline for gaming chat that handles the linguistics of gamer chat such as leetspeak, gaming abbreviations, and censored profanity.

4. A rigorous **five-phase ablation study** demonstrating a cumulative macro-F1 of 0.618 on the shared task.

2 Related Work

Toxicity detection in gaming: Early computational work for classifying game toxicity in chats used crowdsourced moderation signals and rich behavioral features: Blackburn and Kwak (2014) built a large scale corpus of utterances from the game-League of Legends and trained Random Forests classifiers over hundreds of in-game and chat-derived features to predict community level toxicity. Märtens et al. (2015) introduced a lexicon-driven annotation pipeline for the chat for the game-Dota 2 and released a resource called DotAlicious for analysis of utterance-level toxic vs. non-toxic. Stoop et al. (2019) proposed a conversation-aware modeling approach (the HaRe framework), which maintained per-user toxicity estimates and were updated with each new message for detection of real-time harassment in the game-League of Legends. The more recent, Yang et al. (2023) introduced ToxBuster, which conditioned line-level toxicity on the chat history and metadata across several multiplayer titles; in a post-game moderation setting they reported metrics which flagged 82.1% of chat-reported players at a precision of 90.0%, and identifying an additional 6% of toxic players who were not reported by other players.

Multi-class intent and joint token supervision (GameTox): Naseem et al. (2025) released GameTox, 53K utterances from the game-*World of Tanks* with six-way intent labels and token-level slot labels, together with human annotation for classification and LLM verification. On the English-only subset, their baselines indicated that intent level classification is substantially harder than simply using slot filling, the strongest joint model, Joint BERT (Chen et al., 2019), achieved an intent macro-F1 (I-F1) = 0.89, slot macro-F1 (S-F1) = 0.99, and intent accuracy (ICA) = 0.89. The explanation-centric framework called ToXCL proposed in (Hoang et al., 2024) reached an I-F1 = 0.87 and ICA = 0.88. Several of the LLM baselines are below the joint NLU models on this benchmark (e.g., The Gemma-7B model (Gemma Team et al., 2024) I-F1 = 0.74; The Mistral-7B model (Jiang et al., 2023) 0.69; Flan-T5-XL (Chung et al., 2024) 0.68; Llama-2-7B (Touvron et al., 2023) 0.65), this supported the claim that token-level slot supervision helps the models to cope with game-specific obfuscate toxic language.

Offensive language and hate speech (non-gaming benchmarks): The corresponding progress for the toxicity identification on social media included the identification of offensive language (Zampieri et al., 2019), large-scale abusive-behavior characterization (Founta et al., 2018),as well as detection of multi-aspect cyberbullying (Salawu et al., 2021), as well as benchmarks for rationale-grounded hate-speech (Mathew et al., 2021). These resources advanced the toxic language modeling from coarse to fine grain, but the schema for labels as well as domain differ from that of multi-intent game chat, where slang, obfuscation, and community norms dominate (Naseem et al., 2025).

3 Shared Task & Dataset

Task description: The EEUCA 2026 Shared Task on Understanding Toxic Behavior in Gam-

Label	Train	%	Val	%
Non-Toxic	34797	81.00	4349	81.03
Insults & Flaming	5925	13.79	740	13.79
Other Offensive	1874	4.36	234	4.36
Hate & Harassment	279	0.65	34	0.63
Threats	60	0.14	7	0.13
Extremism	24	0.06	3	0.06
Total	**42959**	100	**5367**	100

Table 1: Intent label distribution in our train/validation splits (test size: 5375).

ing Communities (Thapa et al., 2026; Hürriyetoğlu et al., 2026) challenges participants to classify game chat utterances into six toxicity intent categories. The task is framed as a single-label multi-class classification problem, evaluated on the macro-F1 score to equally weight rare and frequent classes to properly account for the class imbalances.

Dataset: The task uses the GameTox dataset (Naseem et al., 2025), comprising about 53,000 utterances from the World of Tanks game chat. The annotation schema follows the toxicity categorization framework established in Bhandari et al. (2023). Each utterance is classified into one of six categories: Non-Toxic, Insults and Flaming, Other Offensive Texts, Hate and Harassment, Threats, and Extremism. The annotations were produced through a human-LLM collaborative process with a three-phase schema achieving a Fleiss' Kappa of 0.91 (Falotico and Quatto, 2015).

Data split: The English-only subset of utterances is split into training (42959), validation (5367), and test (5,375) sets. As shown in Table 1, the dataset exhibits severe class imbalance, with Non-Toxic comprising 81.00% and Extremism only 0.06% of the data.

4 Methodology

Figure 1 illustrates the overall architecture of our proposed TAGA approach. It consists of four components:

4.1 Pre-processing

Gaming chat requires specialized normalization to recover the semantic meaning which is obscured in the content by informal writing conventions. We implement three domain specific methods utterance identifiers for this:

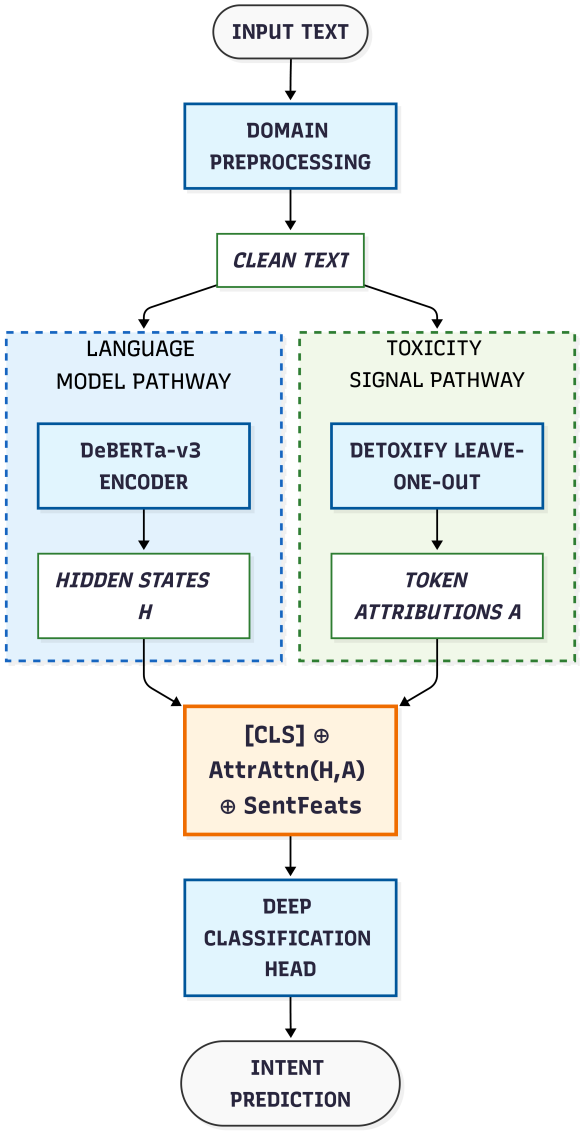

Figure 1: Overview of the TAGA architecture.

Leetspeak Decoding: We decode common character substitutes using numbers found in the dataset like (e.g., "naz1" → "nazi", "b1tch" → "bitch", "h1tler" → "hitler"). These substitutes are generally prevalent in Extremism and Hate categories where users attempt to evade the hate speech filters in various games.

Gaming Abbreviation Expansion: We manually handle 22 gaming-specific abbreviations which are commonly used in World of Tanks and other similar games (e.g., "kys" → "kill yourself", "stfu" → "shut the fuck up", "inting" → "intentional feeding"). This normalization is crucial as many abbreviations carry strong sentiments representing toxicity that would be opaque to the models and not classified as toxic if trained on standard English.

Uncensoring: We use regex-based patterns checks to recover censored words and phrases where characters are replaced with special symbols like (e.g., "f**k" → "fuck", "sh#t" → "shit"). This causes the model to understand and thus leverage the full semantic content of the utterance.

4.2 Token-Level Attribution

A key insight which motivates the use of TAGA is that the contribution of individual tokens to an utterance's toxicity can be estimated by measuring the effect on toxicity on their removal. We compute token-level attributions using the Detoxify toxicity scorer (Hanu and Unitary team, 2020), which is a suite of models trained on the Jigsaw toxicity datasets (Jigsaw/Conversation AI, 2018).

Leave-One-Out Attribution: For each utterance $\mathbf{x} = (w_1, w_2, \ldots, w_n)$, we first obtain baseline toxicity scores considering all tokens $\mathbf{b} \in R^C$ across $C = 4$ channels (toxicity, threat, insult, identity attack) using the Detoxify unbiased model. For utterances where $b_{\text{toxicity}} > \tau$ (we use $\tau = 0.15$), we compute the toxicity attribution of each token w_i by measuring the score drop when that token is removed:

$$a_{i,c} = \max\left(0,\ b_c - s_c(\mathbf{x}_{\setminus i})\right) \tag{1}$$

where $\mathbf{x}_{\setminus i}$ represents the utterance with token w_i removed and $s_c(\cdot)$ returns the score for the channel c. The $\max(0, \cdot)$ clipping makes sure that only tokens that contribute to decreasing toxicity contribute positively to toxicity attributions. This produces an attribution matrix $\mathbf{A} \in R^{n \times C}$ for each utterance.

Efficiency: We minimize attribution computation to the first 30 tokens per sentence and skip non-toxic utterances ($b_{\text{toxicity}} \leq \tau$), setting their attributions to zero which reduces the total number of Detoxify inference calls substantially and retains attributions for all of the remaining toxic content as required.

4.3 Sentence-Level Toxicity Features

In addition to the token-level attributions, we also extract sentence-level features from two Detoxify model variants:

$$\mathbf{f}_{\text{sent}} = [\mathbf{d}_{\text{unbiased}}; \mathbf{d}_{\text{original}}] \in R^{14} \tag{2}$$

where each $\mathbf{d} \in R^7$ contains score for toxicity, severe toxicity, obscenity, threat, insult, identity attack, and sexual explicitness. The dual-models used capture complementary perspectives on toxicity, as the two Detoxify variants were trained on completely different subsets of the Jigsaw data and different debiasing strategies.

4.4 TAGA Model Architecture

Encoder: We implement an encoder, pre-trained DeBERTa-v3-base model (He et al., 2023) as our backbone encoder, which produces contextualized token representations $\mathbf{H} = (\mathbf{h}_1, \ldots, \mathbf{h}_L) \in R^{L \times d}$ where L is the sequence length and $d = 768$ is the hidden dimensions. DeBERTa's disentangled attention mechanism (He et al., 2021) separately encodes the content and position information and provides stronger representations than standard BERT.

Attribution-Guided Attention Pooling: Instead of relying solely on the [CLS] token, we compute a weighted sum of all the token representations and use attention scores that are biased using the token-level attributions. The attention logits are given as:

$$e_i = \underbrace{\mathbf{w}^\top \tanh(\mathbf{W}_a \mathbf{h}_i)}_{\text{content attention}} + \underbrace{g(\mathbf{a}_i)}_{\text{attribution bias}} \tag{3}$$

where $\mathbf{W}_a \in R^{256 \times d}$ and $\mathbf{w} \in R^{256}$ are learnable parameters for content-based attention, $\mathbf{a}_i \in R^C$ is the attribution vector for token i (aligned to subword tokens), and $g : R^C \to R$ is a learned projection:

$$g(\mathbf{a}) = \mathbf{v}^\top \text{GELU}(\mathbf{W}_g \mathbf{a} + \mathbf{b}_g) + b_v \tag{4}$$

with $\mathbf{W}_g \in R^{16 \times C}$. The attribution projection is initialized with small weights (std = 0.02) to ensure that the model relies initially on content-based attention and then gradually learns to incorporate token level attribution signals during training.

The attention-pooled representation is computed as:

$$\mathbf{h}_{\text{attn}} = \sum_{i=1}^{L} \alpha_i \mathbf{h}_i, \quad \alpha_i = \frac{\exp(e_i)}{\sum_j \exp(e_j)} \tag{5}$$

where padding positions are masked with $-\infty$ before applying softmax.

Classification Head: The final representation combines the [CLS] token, the attribution-guided attention pooling, and sentence-level features together into one:

$$\mathbf{z} = [\mathbf{h}_{\text{CLS}}; \mathbf{h}_{\text{attn}}; \mathbf{f}_{\text{sent}}] \in R^{2d+14} \quad (6)$$

This is then sent through a three-layer classification head:

$$\hat{\mathbf{y}} = \mathbf{W}_3 \cdot \text{GELU}(\mathbf{W}_2 \cdot \text{GELU}(\mathbf{W}_1 \mathbf{z})) \quad (7)$$

having hidden dimensions of 512 and 256, and dropout (rate 0.15) being applied after each activation.

4.5 Training

Focal Loss: In order to address the severe class imbalance, we use focal loss (Lin et al., 2017) alongside class-dependent weights:

$$\mathcal{L}_{\text{focal}} = -\sum_{k=1}^{K} w_k (1 - p_k)^{\gamma} y_k \log p_k \quad (8)$$

where $\gamma = 2.0$ is the focusing parameter, y_k is the one-hot target label with label smoothing (Szegedy et al., 2016) ($\epsilon = 0.05$), and $w_k = \sqrt{N/(K \cdot n_k)}$ are class weights derived from the *original* (pre-oversampling) class frequencies which are then clipped to $[1, 5]$.

Auxiliary Token-Level Loss: We implement an auxiliary token-level loss that allows the model to predict token-level toxicity from the hidden representations:

$$\mathcal{L}_{\text{aux}} = \lambda \cdot \beta(t) \cdot \text{MSE}\big(\sigma(\hat{\mathbf{t}}), \text{clamp}(\textstyle\sum_c \mathbf{a}_c, 0, 1)\big) \quad (9)$$

where $\hat{\mathbf{t}}$ are individual token predictions from a lightweight head, $\lambda = 0.02$ controls the auxiliary loss weight, and $\beta(t) = \max(0, 1 - t/T)$ is a linear anneal that reduces the auxiliary supervision over the epochs. This loss causes the encoder to develop token-level toxicity awareness early in training and is gradually relaxed as the main classification objective begins to take precedence.

Total Loss:

$$\mathcal{L} = \mathcal{L}_{\text{focal}} + \mathcal{L}_{\text{aux}} \quad (10)$$

Strategic Oversampling: We apply specific class level oversampling with augmentation to address the extreme class imbalance in the dataset. Target sample counts are set per class (e.g., 2,500 for Other Offensive, 600 for Threats, 300 for Extremism), with duplication up to 15× and text augmentation (word swaps, deletions, duplications, shuffling) for classes with fewer than 100 original samples. Oversampled copies however retain the original Detoxify features while augmented texts receive zero attributions, preventing the model from overfitting to noisy augmented attribution patterns.

Optimization: We use AdamW optimizer with differential learning rates: 1.5×10^{-5} for the encoder and 7.5×10^{-5} (5×) for the classification head and attribution projections. We also use a cosine scheduler with 6% warmup (Loshchilov and Hutter, 2017) that controls the learning rate over 5 epochs of training and has a batch size of 32. We employ mixed-precision training (Micikevicius et al., 2018) and gradient checkpointing (Chen et al., 2016) for memory efficiency on a single NVIDIA T4 GPU.

5 Experiments and Ablation Study

We conducted a rigorous ablation study to isolate the contribution of each component in the system. Starting from a DeBERTa-v3-base baseline, we incrementally added gaming-domain preprocessing, a redesigned loss function, architectural enhancements, and our proposed TAGA feature set incrementally. A final competition configuration (E5) is used to train on the combined train+validation split. With the release of true test labels, all experiments E0–E4 are now evaluated directly on the held-out test split. Performance is reported using Test Macro-F1.

5.1 Results

Table 2 summarizes the cumulative effect of each component on the true test Macro-F1.

5.1.1 Baseline (E0)

Five transformer models are fine-tuned with a single linear classification head on raw, uncleaned game-chat data using standard cross-entropy loss, without class balancing or preprocessing (Table 3). All five models remain below 0.51 macro-F1 which is a disconnect driven by the near-complete failure

Phase	Description	Macro-F1	Accuracy	Precision	Recall
E0	Vanilla DeBERTa-v3, CLS pooling, CE loss, raw text	0.4983	0.8789	0.4731	0.5348
E1	+ L33t decode + slang normalisation + uncensoring	0.5086	0.8798	0.4817	0.5425
E2	+ Focal loss (γ=2) + class weights + label smoothing	0.5114	0.8945	0.5013	0.5245
E3	+ CLS+attention pooling + deeper head + differential LR	0.4959	0.9003	0.4900	0.5047
E4	+ Dual Detoxify (14-d) + attribution + oversample + aux loss	0.5905	0.8915	0.5841	0.6012
E5	Full data (train+val), all E4 components	**0.6186**	**0.8902**	**0.6047**	**0.6497**

Table 2: Incremental test performance across phases.

Model	Macro-F1	Accuracy	Precision	Recall
BERT-base	0.4747	0.8969	0.5079	0.4588
HateBERT	0.4961	0.8966	0.5104	0.4876
ToxicBERT	**0.5044**	0.8997	0.5177	0.4935
RoBERTa-base	0.4976	**0.9018**	**0.5351**	0.4720
DeBERTa-v3-base	0.4983	0.8789	0.4731	**0.5348**

Table 3: Baseline Model Performance on Test Set (Phase E0) where models are arranged in ascending order of parameter size.

on minority classes, where Threats and Extremism yield near-zero F1 across all models. ToxicBERT achieves the best macro-F1 (0.5044), while RoBERTa-base leads in precision (0.5351) and accuracy (0.9018). DeBERTa-v3-base, despite the lowest precision (0.4731), achieves the highest recall (0.5348), reflecting greater sensitivity to minority class instances. Given its disentangled attention mechanism and ELECTRA-style pre-training, DeBERTa-v3-base is selected as the backbone for all subsequent phases.

5.1.2 Gaming-Domain Preprocessing (E1)

Gaming communities employ three main strategies to obscure toxicity in game chat that defeat the baseline tokenizer: Leetspeak ("n4z1" $\rightarrow$ "nazi"), community slang ("noob", "rekt"), and censored profanity ("f***" $\rightarrow$ "fuck"). We evaluate each normalisation step and then run it in combination reporting a test macro-F1 of 0.5086.

5.1.3 Loss Function Redesign (E2)

The six-class distribution is severely imbalanced towards non toxic utterances: Non-toxic (81.0%) versus Extremism (0.06%). The standard cross-entropy loss ignores this imbalance. Thus, we evaluate two specific alternatives:

Focal loss with weights and smoothing (E2b) achieves the highest test F1 of 0.5114, outperforming class-weighted CE (E2a, 0.4961). The combined configuration is nonetheless adopted for its stable training foundation across subsequent phases. Despite improvements, loss reweighting alone cannot overcome extreme data scarcity.

Variant	Configuration	Val F1	Test F1
E2a	Class-weighted CE	0.5061	0.4961
E2b	Focal + weights + smoothing	0.5102	0.5114

Table 4: Loss function ablation across class weighted CE and Focal loss with weights and smoothing.

5.1.4 Architecture Enhancements (E3)

We replace the single CLS token with a learned attention pooling mechanism that combines CLS with all token representations. Thus we hypothesized that toxic signals are concentrated in specific words or threat tokens rather than the global CLS embedding. We also deepened the classification head (Linear$\rightarrow$GELU$\rightarrow$Dropout$\rightarrow$Linear) and applied a $5\times$ differential learning rate multiplier to the head versus the backbone of the model. On the true test set, these changes yield a macro-F1 of 0.4959, a regression from the F1-score in E2. This drop suggests that the deeper head and attention pooling mechanism slightly overfit the training distribution on the unseen test data, confirming that architectural capacity alone is not the primary bottleneck. The performance gap is instead attributed to the absence of explicit toxicity signals, which motivates the feature-level augmentation introduced in E4.

5.1.5 Dual TAGA Feature Set (E4)

Phase 4 introduces our primary contribution to the task: combines the TAGA feature set which comprises (i) dual Detoxify sentence vectors, (ii) token-level attribution scores, (iii) minority-class oversampling, and (iv) an auxiliary toxicity regression loss. We ablate for the Detoxify components in Table 5.

E4a, which introduces sentence-level Detoxify features without oversampling or auxiliary supervision, yields a test macro-F1 of 0.4480, below the E0 baseline (0.4983). This regression indicates that Detoxify features alone are insufficient and may introduce noise without a training objective that directs the model to exploit them; the minority classes remain suppressed under standard cross-entropy. The combination of oversampling and auxiliary loss (E4b) is the decisive factor: even with a single-model 7-d Detoxify vector, test macro-F1 jumps to 0.5887 and Extremism F1 becomes non-zero (0.480) for the first time. The full dual-model configuration (E4) further improves Hate F1 to 0.500 and consolidates Extremism at 0.500. E4 achieves 0.5905, a significant improvement over the E0 baseline.

5.1.6 Final System Configuration (E5)

E5 maximises the available training signal by combining the train and validation splits before fine-tuning, ensuring that every annotated example - including the rarest Extremism (27 samples) and Threats (67 samples) - contributes to the final model. Training on the full labelled data yields a test macro-F1 of **0.618**, improving over E4 and our best submitted result.

5.2 Per-Class Analysis

Table 6 traces per-class F1 across all phases. Non-toxic and Insults remain stable across all phases, confirming that majority classes are well-captured from the baseline. The Hate class improves consistently from 0.476 (E0) to 0.542 (E2), driven by slang normalisation in E1 and loss redesign in E2, before settling at 0.500 in E4. Threats shows rising sharply from 0.338 (E0) to 0.410 (E1) due to L33t-speak decoding, dipping through E2-E3, and recovering to 0.420 in E5 with oversampling and auxiliary loss. Extremism remains zero across E0–E3 and first becomes non-zero in E4 (0.500), rising further to **0.667** in E5 the single largest per-class gain across all phases, directly attributable to the TAGA oversampling and token attribution strategy providing sufficient training signal for this 24-sample minority class.

5.3 Error Analysis

We believe that documenting negative results and error analyses is as valuable as reporting performance gains. Therefore, we detail both the successes of the TAGA system and the surprising failure modes that emerged across our five-phase ablation.

Architecture regression in E3. A counterintuitive finding is that the introduction of attention pooling and a deeper classification head in E3 *decreased* test macro-F1 by 1.55 pp relative to E2, despite these components being theoretically well-motivated. We hypothesize that the increased capacity of the head, combined with the absence of explicit toxicity signals, caused the model to overfit to the majority-class distribution of the training data. This result suggests that architectural complexity without complementary feature-level inductive bias is insufficient, and potentially harmful, for severely imbalanced datasets.

Detoxify features without oversampling are counterproductive. E4a, which adds sentence-level Detoxify features without oversampling or auxiliary supervision, produces a test macro-F1 of 0.4480 below the E0 baseline of 0.4983. This negative result reveals that injecting external toxicity signals into a model trained under standard cross-entropy on an imbalanced dataset can actively degrade minority class performance. The Detoxify features introduce a richer signal space that the model cannot exploit without a complementary training objective that forces recovery of minority classes. This finding motivates the joint introduction of oversampling and auxiliary loss in E4b, which together produce the decisive performance jump.

Confusion between Extremism and Hate. Even in our best configuration (E5), the model achieves only 0.469 on the Hate class despite achieving 0.667 on Extremism. Manual inspection of misclassified utterances reveals that group-targeted language involving political or ethnic references is frequently confused between the two categories, as the distinction requires contextual world knowledge that extends beyond the surface form of the utterance. This is an inherent limitation of the classification of the level of utterance without the

Variant	Components	Test F1	Threats	Extremism	Hate
E4a	7-d unbiased (Sent-level only) (no oversample, no aux)	0.4480	0.2500	0.000	0.3950
E4b	7-d unbiased + oversample + aux	0.5887	0.4200	0.4800	0.4750
E4	Dual TAGA (14-d + attr + oversample + aux)	**0.5905**	0.3880	**0.5000**	**0.5000**

Table 5: Detoxify component ablation showing the impact of sentence-level features, oversampling, and token attribution on Test F1 across the rarest toxicity classes.

Phase	Non-toxic	Insults	Other	Hate	Threats	Extremism	Macro-F1
E0 Baseline	0.952	0.768	0.456	0.476	0.338	0.000	0.498
E1 Preprocessing	0.951	0.764	0.418	0.509	0.410	0.000	0.508
E2 Loss redesign	0.946	0.762	0.460	0.542	0.350	0.000	0.511
E3 Architecture	0.948	0.765	0.442	0.495	0.325	0.000	0.495
E4 Dual TAGA	0.948	0.762	0.445	0.500	0.388	0.500	0.590
E5 Full TAGA	0.947	0.763	0.436	0.469	0.425	**0.667**	**0.618**

Table 6: Per-class Test F1 across ablation phases.

context of the discourse.

6 Conclusion

We presented our TAGA architecture, which is a submission to the EEUCA 2026 Shared Task on Understanding Toxic Behavior in Gaming Communities (Thapa et al., 2026). Through a systematic and rigorous five-phase ablation, we demonstrated that each component contributes meaningfully to the final result.

Our work demonstrates that extracting explicit toxicity attribution from pre-trained scorers can serve as an effective and meaningful inductive bias for attention-based models, providing a principled way to incorporate domain knowledge without manual human token-level annotation. Future work could explore extending TAGA to joint intent-slot classification, applying the attribution mechanism to other toxicity detection domains, and investigating gradient-based attribution methods to reduce the $O(n)$ preprocessing cost of leave-one-out perturbation.

Limitations

Our work has several limitations. First, the leave-one-out attribution computation requires $O(n)$ time forward passes through the Detoxify model per toxic utterance, making it expensive to run at scale. Gradient-based attribution methods could provide a more efficient alternative. Second, our approach is evaluated only on a single dataset (GameTox) from one game (World of Tanks), and generalization to other gaming communities with different linguistic norms and game talk remains to be validated. Third, the extreme rarity of certain categories (27 Extremism, 67 Threats samples total) makes robust evaluation statistically challenging, and results on these classes should be interpreted with appropriate uncertainty. Finally, our preprocessing pipeline relies on manually curated lexicons for leetspeak and gaming abbreviations, which may not generalize to evolving gaming slang and may require manual updating.

Ethical Considerations

This work involves the processing and classification of toxic, hateful, and extremist language from real-world gaming chat. While our goal is to advance automated moderation to protect players from harm, several ethical considerations warrant attention. First, the GameTox dataset contains genuine instances of hate speech, threats, and extremist content; access to and use of such data should be restricted to research purposes and handled responsibly to avoid amplifying harmful content. Second, automated toxicity classifiers are inherently imperfect, our system achieves a test macro-F1 of 0.618 and deploying such a system in a production moderation pipeline without human intervention risks both false positives, which may unfairly penalize legitimate players using game-specific language, and false negatives, which may allow genuinely harmful content to go unmoderated. Third, the preprocessing lexicons and Detoxify scorer used in TAGA reflect toxicity norms primarily from English-language Western gaming communities; application to other languages, cultures, or game genres may introduce cultural bias and should be validated independently before deployment. Finally, we acknowledge that systems trained to de-

tect extremist and hateful content could, if misused, be repurposed to identify and target individuals who express such views rather than to protect potential victims, and we strongly discourage any such application of this work.

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.

Jeremy Blackburn and Haewoon Kwak. 2014. Stfu noob! predicting crowdsourced decisions on toxic behavior in online games. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 877–888.

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

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. 2016. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Bruno Mendes da Silva, Mirian Tavares, Filipa Cerol, Susana Mendes da Silva, Paulo Falcão, and Beatriz Isca Alves. 2020. Playing against hate speech – how teens see hate speech in video games and online gaming communities. volume 3, pages 34–52.

Rosa Falotico and Piero Quatto. 2015. Fleiss' kappa statistic without paradoxes. *Quality & Quantity*, 49:463–470.

Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and 1 others. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Laura Hanu and Unitary team. 2020. Detoxify. https://github.com/unitaryai/detoxify.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *International Conference on Learning Representations*.

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. *arXiv 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)*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Jigsaw/Conversation AI. 2018. Toxic comment classification challenge. https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge.

Haewoon Kwak, Jeremy Blackburn, and Seungyeop Han. 2015. Exploring cyberbullying and other toxic behavior in team competition online games. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3739–3748.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988.

Ilya Loshchilov and Frank Hutter. 2017. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*.

Marcus Märtens, Siqi Shen, Alexandru Iosup, and Fernando Kuipers. 2015. Toxicity detection in multiplayer online games. In *2015 International Workshop on Network and Systems Support for Games (NetGames)*, pages 1–6.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14867–14875.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh

Venkatesh, and Hao Wu. 2018. Mixed precision training. In *International Conference on Learning Representations*.

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.

Anil Singh Parihar, Surendrabikram Thapa, and Sushruti Mishra. 2021. Hate speech detection using natural language processing: Applications and challenges. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1302–1308. IEEE.

Semiu Salawu, Jo Lumsden, and Yulan He. 2021. A large-scale english multi-label twitter dataset for cyberbullying and online abuse detection. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 146–156.

Wessel Stoop, Florian Kunneman, Antal van den Bosch, and Ben Miller. 2019. Detecting harassment in real-time as conversations develop. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 19–24.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

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)*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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*, pages 9894–9906, Singapore. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the type and target of offensive posts in social media. *arXiv preprint arXiv:1902.09666*.

A Hyperparameter Configuration

Table 7 lists the full hyperparameter configuration for the TAGA model.

Hyperparameter	Value
Encoder	DeBERTa-v3-base
Max sequence length	128
Batch size	32
Encoder learning rate	1.5×10^{-5}
Head learning rate multiplier	$5\times$
Weight decay	0.01
Dropout rate	0.15
Epochs	5
Warmup ratio	6%
Max gradient norm	1.0
Focal loss γ	2.0
Label smoothing ϵ	0.05
Auxiliary loss weight λ	0.02
Attribution toxicity threshold τ	0.15
Max attribution tokens	30
Attribution channels	4
Detoxify models	unbiased, original
Sentence features dim	14
Oversampling targets	
Other Offensive	2,500
Hate and Harassment	1,500
Threats	600
Extremism	300

Table 7: Full hyperparameter configuration for TAGA.

B Model Architecture Details

The TAGA model consists of the following components:

- **Encoder:** DeBERTa-v3-base (184M parameters) with gradient checkpointing enabled.
- **Attention projection:** Linear(d, 256) $\rightarrow$ Tanh $\rightarrow$ Linear(256, 1).
- **Attribution projection:** Linear(C, 16) $\rightarrow$ GELU $\rightarrow$ Linear(16, 1), initialized with $\mathcal{N}(0, 0.02)$.
- **Token head (auxiliary):** Linear(d, 64) $\rightarrow$ GELU $\rightarrow$ Dropout(0.15) $\rightarrow$ Linear(64, 1).
- **Classification head:** Linear($2d + 14$, 512) $\rightarrow$ GELU $\rightarrow$ Dropout(0.15) $\rightarrow$ Linear(512, 256) $\rightarrow$ GELU $\rightarrow$ Dropout(0.15) $\rightarrow$ Linear(256, 6).