wangkongqiang@EEUCA 2026: Understanding Toxic Behavioral Intent in Gaming Chat Logs

Kongqiang Wang
School of Information Science and Engineering, Yunnan University, Yunnan Baiyao Street, 650500, Kunming, Yunnan, China.
wangkongqiang60@gmail.com

Peng Zhang
School of Information Science and Engineering, Yunnan University, Yunnan Baiyao Street, 650500, Kunming, Yunnan, China.
zpp1219@gmail.com

Qingli Tan
College of Ecology and Environment, Yunnan University, Yunnan Baiyao Street, 650500, Kunming, Yunnan, China.
tanqingli@stu.ynu.edu.cn

Abstract

Our team was interested in content classification and labeling from toxicity detection of gaming chat logs in online gaming communities. We joined the shared task on Understanding Toxic Behavioral Intent in Gaming Chat Logs@EEUCA with ACL 2026. In this task, our goal is to assign a content classification label to player's utterance (e.g., Hate and Harassment, Threats, Non-toxic). The objective is to develop systems that can classify the intent of a player's utterance. The dataset for this task will have five labels: Non-toxic (0), Insults and Flaming (1), Other Offensive Texts (2), Hate and Harassment (3), Threats (4) and Extremism (5). The performance will be ranked by F1-score (Macro). The task utilizes 53,000 game chat utterances from *World of Tanks*. Our group used a supervised learning method on multiple pre-trained models and finetuning Qwen2 LLMs. The best result on the test set for shared task were Macro F1 score of 0.5776, Accuracy 0.9075, Precision (Macro) 0.6847, and Recall (Macro) 0.5343 from finetuning qwen2_7B LLM method, ranking 8th among all teams. The complete code of this entire project can be found at our GitHub address[1].

1 Introduction

First of all, let's introduce the overview of shared task on Understanding Toxic Behavioral Intent in Gaming Chat Logs@EEUCA with ACL 2026 (Thapa et al., 2026). The prevalence of toxic behavior in online gaming communities necessitates robust detection methods to ensure user safety. This shared task focuses on detecting toxicity in game chat logs, specifically using the *GameTox* dataset, which captures the complex relationship between user intent and specific linguistic features. In this context, the distinction between Toxic and Non-toxic becomes blurred, as gaming chat logs straddle the line between satire and offense, challenging researchers and platforms alike to navigate the complexities of online content moderation. As one label generally fails to encompass multiple aspects of linguistics, this shared task classifies gaming chat logs on five aspects: Non-toxic (0), Insults and Flaming (1), Other Offensive Texts (2), Hate and Harassment (3), Threats (4) and Extremism (5).

The objective is to develop systems that can classify the intent of a player's utterance (e.g., Hate and Harassment, Threats, Non-toxic). The task utilizes 53,000 game chat utterances from *World of Tanks*. This dataset has been published in NAACL 2025 (Naseem et al., 2025).

2 Background

2.1 Toxicity detection in online games

The internet has become a common platform for everyone to share their ideas and opinions. The user has freedom to post whatever he/she likes in social networking and blogging sites. However, sometimes the content when directed towards certain group of individuals with an intention to incite hate or discrimination, causes a turmoil in the society. Such content is known as hate speech. Hate speech (Bhandari et al., 2023) can be a serious problem to peace and harmony in the society. There are instances where hate speech have led to social unrest and extremism. Thus, hate speech in the internet needs to be monitored (Parihar et al., 2021). In this context, researchers have proposed various frameworks and datasets for automated toxicity detection in online games. (Blackburn and Kwak, 2014) utilized crowdsourced in-game user reports from League of Legends (LoL) for toxic

[1] https://github.com/WangKongQiang/EEUCA2026_Understanding_Toxic_Behavioral_Intent_in_Gaming_Chat_Logs

Proceedings of the 9th Workshop on Event Extraction and Understanding: Challenges and Applications, pages 104–111

behavior detection by extracting 534 features from in-game performance, user reports, and chat logs and employed the Random Forest Classifier for toxicity detection. (Stoop et al., 2019) used a similar approach for data collection and introduced the RNN-based HaRe framework that tracked toxicity estimates for each user individually, updated the estimate with every new utterance, concatenated all of the utterances of each user, and classified the combined text. (Märtens et al., 2015) proposed a novel lexicon-based annotation strategy for game chat toxicity detection to devise the DotAlicious dataset consisting of chat replays from 12,923 Defense of the Ancients (DOTA) matches.

2.2 Toxicity and Hate speech datasets

Detection of hate speech and toxicity in online environments has seen significant progress in recent years. (Oz et al., 2023) aimed to explore the perceptions, concerns, and strategies of LGBTQ social media activists in Turkey. Through semi-structured interviews with 20 LGBTQ social media activists, This study investigated how they navigate cultural and political challenges and utilize social media for activism purposes. (Thapa et al., 2024) addressed the need for effective hate speech moderation in contemporary digital discourse, the multimodal hate speech event detection shared task (Thapa et al., 2023) made its debut at russia-ukraine crisis period. (Qian et al., 2019) introduced two labeled hate speech datasets collected from Reddit (22k comments) and Gab (33k comments) containing manually-written intervention responses. (Wijesiriwardene et al., 2020) focused on toxic behaviors among youngsters and introduced ALONE, a dataset for toxic behavior detection among adolescents on Twitter, consisting of 16,901 tweets in 688 interactions and labeled for toxic vs non-toxic classes. (Founta et al., 2018) analyzed abusive behavior on Twitter by releasing a dataset of 80,000 tweets annotated for seven labels: offensive, abusive, hate speech labels, aggressive, cyberbullying, spam, and normal. (Mathew et al., 2020) introduced HateXplain, a dataset for explainable hate speech detection, consisting of 20,148 posts collected from Twitter and Gab annotated for three classes: hate, offensive, and normal, alongside target communities within hate. They further annotated the sections of the post that guide the labeling rationale. (Zampieri et al., 2019) released an offensive language detection dataset comprising 14,100 tweets categorizing offensive language and its targets, consisting of offensiveness detection with three target classes: Individual, Group, and Other. To discern multiple aspects within cyberbullying, (Salawu et al., 2021) curated an extensive dataset for cyberbullying detection comprising 62,587 tweets annotated for multiple aspects including Bullying, Profanity, Sarcasm, Threat, and Spam.

3 Dataset

In this section, we describe various aspects of task dataset including data collection, utterance annotation, and dataset statistics. Task dataset comprises 42,963 text utterance that encompass different intent content relevant to the chat recordings from the game *World Of Tanks*. Organizers collected 53,000 utterances from the WoT Record database, which stores chat recordings from the game *World Of Tanks*. Among these utterances, 42,963 samples contained only English text, and the rest were in other languages or a code-mixed format. The 42,963 English utterances were annotated for intent, and all samples were annotated for slot filling by converting the code-mixed samples to English by using Google Translate[2]. Organizers converted all text to lowercase to ensure uniformity. They removed all duplicated text from the corpus, which may otherwise create biases. Further, they removed all user identifiers such as usernames and gamer tags to preserve the privacy of players.

3.1 Utterance Annotation

Each utterance was labeled to one of 6 labels:Non-toxic (0), Insults and Flaming (1), Other Offensive Texts (2), Hate and Harassment (3), Threats (4), and Extremism (5). Non-toxic if toxicity was not present and one of the five toxicity labels if toxicity was present. Utterance annotation for each label are mentioned below.

Hate and Harassment: Utterances with the presence of identity-based hate or harassment (e.g., racism, sexism, homophobia).

Threats: Utterances with threats of violence, physical harm to another player, employee, or property, terrorism, or releasing a player's real-world personal information (e.g., doxing).

Extremism: Utterances with extremist views (e.g., white supremacy), attempts to groom or recruit for an extremist group, or repeated sharing of

[2]https://translate.google.com

Figure 1: Wordcloud of words in each intent label.

political or religious beliefs.

Insults and Flaming: Insults or attacks on another player or team (not based on player or team's real or perceived identity)

Other Offensive Texts: Any message not covered in the aforementioned categories that is offensive or harms a player's reasonable enjoyment of the game.

Non-Toxic: Utterances without any toxicity.

3.2 Dataset Statistics

Table 1: Dataset statistics for GameTox. The data consists of 42,963 samples for player's utterance toxicity detection task in online gaming communities.

Task	Label	#Samples	%
Intent Classification	Non-Toxic	34679	80.71
	Insults and Flaming	6049	14.07
	Other Offensive Texts	1885	4.38
	Hate and Harassment	274	0.63
	Threats	53	0.12
	Extremism	23	0.053
Task	**Token**	**%**	
Slot Classification	Other	67.17	
	Verb	15.51	
	Game Slang	7.72	
	Toxic	9.59	

Table 1 (*Upper*) provides the class distribution of intent across the 42,963 English utterances, and Table 1 (*Down*) provides the slot filling distribution across all utterances. Most utterances are non-toxic in nature and a notable data imbalance is present. However, this is in line with real world data distributions, where extremely toxic labels such as Hate and Harassment, Threats, and Extremism are often moderated or automatically suppressed. Figure 1 illustrates the word cloud for all intent labels.

4 System Overview

4.1 Fine-tuning Pre-trained Models

Introduction. In recent years, with the rapid development of deep learning technology, large-scale pre-trained models have achieved remarkable results in fields such as natural language processing, computer vision, and multimodal learning. Compared with traditional models trained from scratch, pre-trained models can learn rich semantic representations and general knowledge by pre-training on large-scale general corpora, thereby significantly improving the performance and training efficiency of downstream tasks. However, the knowledge learned by pre-trained models on general corpora often has strong generalization, while specific tasks usually have obvious domain characteristics. Therefore, directly applying pre-trained models to downstream tasks often fails to achieve the best results. To solve this problem, researchers usually adopt the fine-tuning strategy, that is, on the basis of the pre-trained model, further optimize the model parameters using the data of specific tasks, so that it can better adapt to the target task. In this study, to enhance the model's performance in the task of toxic behavioral intent analysis, a pre-trained language model was adopted as the base model and fine-tuned in combination with specific task data (*GameTox* dataset), enabling the model to effectively learn the semantic relationship between gaming chat utterance expressions and their underlying intents.

The classifier in the pre-trained model uses a transformer based classifier. The specific pre-trained models of the classifier are shown in the Table 2.

Table 2: pre-trained model classifier structure for gaming chat content classification.

Model	Batch Size	Num Epochs	Learning Rate
albert/albert-base-v2	32	5	2e-5
google-bert/bert-base-uncased	32	5	2e-5
nghuyong/ernie-2.0-large-en	32	5	1e-5
FacebookAI/roberta-large-mnli	32	5	2e-5
cambridgeltl/trans-encoder-bi-simcse-roberta-large	32	5	1e-5

The Principle of Fine-tuning Pre-trained Models. Pre-trained models typically use large-scale corpora for self-supervised learning, such as tasks like language model prediction, masked language modeling, or autoregressive modeling, thereby learning common language representations. After pre-training, the model parameters already contain a large amount of language knowledge and semantic information. Therefore, in downstream

tasks, only a small amount of labeled data is needed to achieve good performance.

The basic idea of fine-tuning is to introduce supervisory signals from downstream tasks on the basis of the parameters of the pre-trained model and further optimize the model parameters through the gradient descent algorithm. Let the parameters of the pre-trained model be θ, Given the downstream task training dataset $\mathcal{D} = (x_i, y_i)_{i=1}^{N}$, Where x_i represents the input text and y_i represents the corresponding label, then the model training objective can be expressed as:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell\left(f(x_i; \theta), y_i\right) \tag{1}$$

Here, $f(\cdot)$ represents the model prediction function, and $\ell(\cdot)$ is the loss function (such as cross-entropy loss). By minimizing this loss function, the model can gradually adapt to the data distribution of a specific task, thereby enhancing the prediction performance.

In practical applications, fine-tuning typically includes the following two methods:

- Full Fine-tuning: Update all the parameters of the pre-trained model to enable it to fully adapt to the target task.
- Parameter-efficient Fine-tuning: Only update some parameters or introduce additional lightweight modules to reduce training costs, such as Adapter, LoRA and other methods.

In this study, based on the characteristics of the task and the availability of computing resources, the pre-trained model was trained using a parameter-efficient fine-tuning strategy.

Input Data Construction. First of all, the original data needs to be converted into an input format that the model can handle. For text tasks, the following steps are usually required:

- Text Cleaning and Preprocessing: Remove irrelevant symbols or abnormal characters;
- Word Segmentation and Encoding: Use the tokenizer corresponding to the pre-trained model to convert the text into a token sequence;
- Input Sequence Construction: Ultimately, the input text will be represented as a sequence of token ids and input into the pre-trained model for feature encoding.

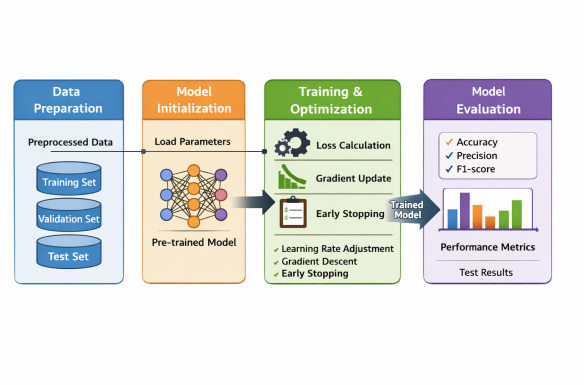

Figure 2: The framework diagram of the fine-tuning pre-trained classification model.

Task Structure Design. During the fine-tuning process, it is necessary to design the corresponding prediction structure based on the specific task. For instance, in the task of toxic behavioral intent analysis in gaming chat logs, the model needs to simultaneously identify toxic behavioral intent categories as well as the corresponding player's utterance information. Therefore, a model is usually composed of the following parts:

- Pre-trained Encoding Layer: Used for extracting semantic representations of text;
- Task-specific Layer: For example, the classification layer or the sequence labeling layer;
- Output Layer: Generate the final prediction result.

By adding task-related structures at the top of the pre-trained model, the model's adaptability to specific tasks can be effectively enhanced.

The overall architecture diagram of the fine-tuing pre-trained model is shown in the Figure 2.

Design of Loss Function. During the training process, it is necessary to select an appropriate loss function based on the type of task. For classification tasks, the cross-entropy loss function is usually adopted.

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \log(p_i) \tag{2}$$

Here, C represents the number of categories, y_i is the true label, and p_i is the model prediction probability. By minimizing the loss function, the consistency between the model's prediction results and the true labels can be gradually improved.

4.2 Hard Voting Mechanism

The hard voting mechanism is a common model fusion strategy in ensemble learning, mainly used for classification tasks. Its basic idea is: multiple base learners make predictions on the same sample respectively, and then determine the final prediction category through majority voting.

The Principle of Hard Voting Mechanism. Assume that the ensemble model consists of M base classifiers: $h_1(x), h_2(x), \ldots, h_M(x)$, where $h_i(x)$ denotes the prediction of the i-th classifier for input sample x. The final prediction of the hard voting ensemble is determined by majority voting:

$$\hat{y} = \arg\max_{c \in C} \sum_{i=1}^{M} I(h_i(x) = c) \tag{3}$$

where C represents the set of all possible classes and $I(\cdot)$ is an indicator function defined as:

$$I(h_i(x) = c) = \begin{cases} 1, & \text{if } h_i(x) = c \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

In a weighted hard voting scheme, each classifier is assigned a weight w_i, and the final prediction can be written as:

$$\hat{y} = \arg\max_{c \in C} \sum_{i=1}^{M} w_i \cdot I(h_i(x) = c) \tag{5}$$

In our experiment, the weights of each classifier were the same.

4.3 Fine-tuning of the Qwen2 Large Language Model (LLM)

Qwen2 is an open-source large language model (LLM) developed by the Tongyi Qianwen team and created by Alibaba Cloud's Tongyi Lab. Using Qwen2 as the base large language model (LLM) and achieving high-accuracy text classification through instruction fine-tuning is an introductory task for learning the fine-tuning of large language models (LLMs).

Instruction fine-tuning is a process of further training an LLMs on a dataset composed of (instruction, input, output) combine pairs. Among them, the instructions represent the human instructions of the model, the input represent the raw data content from specific dataset, and the output represents the expected output that follows the instructions. This process helps bridge the gap between the next word prediction target of LLMs and the goal of users to have LLMs follow human instructions.

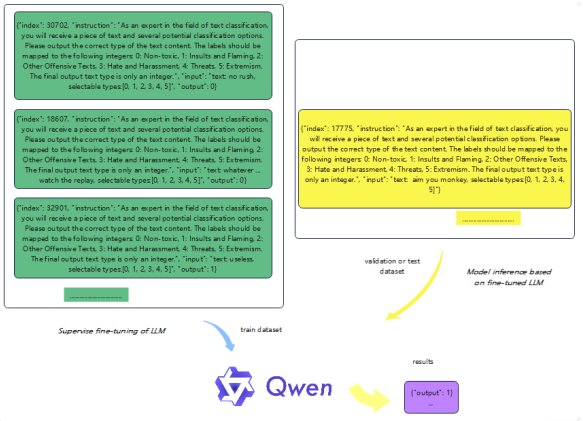

Figure 3: The framework diagram of the fine-tuning qwen2 classification LLM and model inference.

In this toxic behavioral intent classification task in gaming chat logs, we will use the Qwen2-1.5B-Instruct and Qwen2-7B-Instruct model to perform instruction fine-tuning tasks on the dataset, while using SwanLab [3] for monitoring and visualization. The following presents three demonstration formatted data samples for fine-tuning LLM data in the train dataset. Our training task is to ensure that the fine-tuned large language model (LLM) can predict the correct output based on the prompt words composed of text and selectable types.

The complete process of fine-tuning the Qwen2 large language model (LLM) using the train dataset and conducting model inference on validation or test dataset with the fine-tuned large language model (LLM), as shown in Figure 3.

5 Results and Analysis

For results obtained by our fine-tuning pre-trained models and fine-tuning Qwen2 LLM methods on the validation dataset and test dataset are shown in Table 3 and Table 4 respectively. roberta-RNN indicates the addition of a layer of recurrent neural network (RNN) after *FacebookAI/roberta-large-mnli* model, which is the LSTM layer. roberta-cnn indicates adding a convolutional neural network (CNN) layer after *FacebookAI/roberta-large-mnli*, which is the Conv2d layer. bagging refers to the model ensemble decision-making method that employs hard voting mechanism in the four models: roberta, roberta-RNN, simcse-roberta-lstm, roberta-lstm-gru.

[3] https://swanlab.cn

Table 3: The results obtained by our fine-tuning pre-trained models and fine-tuning Qwen2 LLM methods for toxic behavioral intent classification task on the validation dataset.

Pre-trained Model	**Recall (Macro)**	**Precision (Macro)**	**F1 (Macro)**	**Accuracy**
albert/albert-base-v2	0.3416	0.3578	0.348	0.8897
google-bert/bert-base-uncased	0.4118	0.4386	0.4212	0.8975
nghuyong/ernie-2.0-large-en	0.357	0.3737	0.3638	0.9005
FacebookAI/roberta-large-mnli	0.3997	0.4355	0.4131	0.9001
cambridgeltl/trans-encoder-bi-simcse-roberta-large	0.3593	0.4522	0.3702	0.8985
roberta-RNN	0.3624	0.3704	0.366	0.8999
roberta-cnn	0.3502	0.3684	0.3584	0.8975
roberta-lstm-gru	0.3605	0.3638	0.3619	0.8985
simcse-roberta-lstm	0.4047	0.4771	0.4236	0.8985
bagging	0.3562	0.3745	0.3639	0.9018
Large Language Model	**Recall (Macro)**	**Precision (Macro)**	**F1 (Macro)**	**Accuracy**
qwen2_1.5B	-	-	-	-
qwen2_7B	-	-	-	-

Table 4: The results obtained by our fine-tuning pre-trained models and fine-tuning Qwen2 LLM methods for toxic behavioral intent classification task on the test set.

Pre-trained Model	**Recall (Macro)**	**Precision (Macro)**	**F1 (Macro)**	**Accuracy**
albert/albert-base-v2	0.3441	0.3722	0.3553	0.8973
google-bert/bert-base-uncased	0.4109	0.4365	0.4205	0.8928
nghuyong/ernie-2.0-large-en	0.3454	0.3634	0.3533	0.899
FacebookAI/roberta-large-mnli	0.3853	0.4233	0.3992	0.9014
cambridgeltl/trans-encoder-bi-simcse-roberta-large	0.3596	0.5416	0.3785	0.9025
roberta-RNN	0.3547	0.3685	0.3609	0.9003
roberta-cnn	0.3496	0.3699	0.3587	0.9005
roberta-lstm-gru	0.3548	0.3624	0.3584	0.8997
simcse-roberta-lstm	0.4	0.4761	0.4234	0.9031
bagging	0.358	0.5424	0.372	0.9038
Large Language Model	**Recall (Macro)**	**Precision (Macro)**	**F1 (Macro)**	**Accuracy**
qwen2_1.5B	0.4429	0.4646	0.4525	0.9057
qwen2_7B	0.5343	0.6847	0.5776	0.9075

6 Discussion

For shared task on Understanding Toxic Behavioral Intent in Gaming Chat Logs@EEUCA with ACL 2026, we referred to the relevant tasks of CASE 2025 (Hurriyetoglu et al., 2025), CASE 2024 (Thapa et al., 2024) and CASE 2023 (Thapa et al., 2023) shared tasks on multimodal hate speech detection and derived our own method. Although the effect of the experiment needs to be strengthened. However, these contents and ideas have given us a lot of inspiration. Toxic behavioral intent content analysis is a longstanding tradition of the EEUCA workshop series. We believe that with our further research and more detailed optimization on training of the model, we will achieve even greater success in future competitions.

7 Conclusion

We employed multiple methods in detection of toxic behavioral intent in gaming chat logs, which respectively involved the transformer pre-trained models and Qwen2 LLM. Our final leaderboard is shown in the Table 5. The best result of this task was achieved by fine-tuning the Qwen2_7B large language model (LLM) and conducting inference on the test set.

Table 5: The final leaderboard of shared task on Understanding Toxic Behavioral Intent in Gaming Chat Logs@EEUCA with ACL 2026.

#	Username	Recall (Macro)	Precision (Macro)	F1 (Macro)	Accuracy
1	syuhhh-637901	0.7986	0.64	0.7041	0.8982
2	ramihai-572801	0.6846	0.6636	0.6725	0.8992
3	anmolguragain-637916	0.6601	0.6334	0.6441	0.9062
4	srikarkashyap-635409	0.6814	0.5864	0.6234	0.88
5	akshyatshah-636282	0.6497	0.6047	0.6186	0.8902
6	yinloonkhor-636292	0.5946	0.6098	0.5932	0.8925
7	shrinep-637207	0.659	0.554	0.5883	0.9031
8	**wangkongqiang-504685**	**0.5343**	**0.6847**	**0.5776**	**0.9075**
9	dkhonker-536426	0.5815	0.6214	0.5749	0.8865
10	_alexcristea-610819	0.5754	0.5652	0.5632	0.8733
11	akking-609884	0.6002	0.5239	0.5563	0.8876
12	rukesh-shrestha-503743	0.5557	0.5599	0.5539	0.8932
13	nepalshr-637149	0.6476	0.5201	0.5512	0.893
14	merrli-510969	0.6137	0.4798	0.5302	0.8603
15	xiaotian-518453	0.5291	0.5402	0.5301	0.8969
16	runick_allure-508659	0.5328	0.5441	0.5281	0.8772
17	rohanmainali-491803	0.5221	0.5192	0.5192	0.8893
18	linus-636500	0.5134	0.5191	0.5104	0.8716
19	xiaoyu666-603164	0.4884	0.5156	0.4984	0.8951
20	havnis-610798	0.5083	0.4766	0.4895	0.8794
21	giris-585517	0.4895	0.5081	0.4878	0.8964
22	shashi_sah-637803	0.4774	0.5001	0.4869	0.8999
23	wjyyyy-609715	0.4732	0.4962	0.4774	0.8953
24	justdoi-613394	0.5071	0.4487	0.4737	0.8973
25	barkion-610469	0.5002	0.4538	0.4726	0.8781
26	mestecha-623302	0.495	0.4763	0.4686	0.8927
27	binayakkarki-589485	0.4688	0.4647	0.4645	0.8921
28	syhhh-610772	0.5659	0.4198	0.4641	0.7792
29	exterio-610602	0.5084	0.4205	0.4491	0.8443
30	zmin123-554678	0.4568	0.4646	0.4487	0.8506
31	aryankafle-524077	0.4373	0.449	0.4421	0.8962
32	liutianyong-605718	0.4219	0.4701	0.4413	0.9036
33	quasar-501127	0.5357	0.3943	0.4169	0.6471
34	alexandru412-511289	0.6432	0.3315	0.3783	0.7068
35	wenbin-520996	0.1653	0.1629	0.1558	0.7784

8 Limitations of the Work

we are interested in learning about LLMs in computational social science (Thapa et al., 2025), our paper mainly focuses on making discussions on player's utterance for this toxic behavioral intent classification task. This is because we are quite interested in and good at identifying hate and offense categories in the text (Parihar et al., 2021). Due to our lack of utilization of context features, we are unable to make good use of the utterance content in the train dataset of this sharing task. we have chosen the 7B version of Qwen2 due to the limited computing resources. If we could use a larger language model with more parameters, we would achieve better prediction results. These are all our future tasks.

Acknowledgments

We are very grateful to the organizers of the Shared Task on Understanding Toxic Behavioral Intent in Gaming Chat Logs@EEUCA with ACL 2026 (Hürriyetoğlu et al., 2026) and the School of Information Science and Engineering of Yunnan University for providing the experimental environment and equipment.

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*, page 877–888.

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. *Proceedings of the International AAAI Conference on Web and Social Media*, 12(1).

Ali Hurriyetoglu, Surendrabikram Thapa, Hristo Tanev, and Surabhi Adhikari. 2025. Findings and insights from the 8th workshop on challenges and applications of automated extraction of socio-political events from text. In *Proceedings of the 8th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Texts*, pages 1–5, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

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)*.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2020. Hatexplain: A benchmark dataset for explainable hate speech detection. In *AAAI Conference on Artificial Intelligence*.

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)*, page 1–6. IEEE.

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.

Mustafa Oz, Akan Yanik, and Mikail Batu. 2023. Under the shadow of culture and politics: Understanding lgbtq social media activists' perceptions, concerns, and strategies. *Social Media + Society*, 9(3).

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.

Jing Qian, Anna Bethke, Yinyin Liu, Elizabeth Belding, and William Yang Wang. 2019. A benchmark dataset for learning to intervene in online hate speech.

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, Online. Association for Computational Linguistics.

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.

Surendrabikram Thapa, Farhan Ahmad Jafri, Ali Hürriyetoğlu, Francielle Vargas, Roy Ka Wei Lee, and Usman Naseem. 2023. Multimodal hate speech event detection-shared task 4. In *CASE 2023-Proceedings of the 6th Workshop on Challenges and Applications of Automated Extraction of Socio-Political Events from Text, associated with 14th International Conference on Recent Advances in Natural Language Processing, RANLP 2023*, pages 151–159. Association for Computational Linguistics.

Surendrabikram Thapa, Kritesh Rauniyar, Farhan Ahmad Jafri, Hariram Veeramani, Raghav Jain, Sandesh Jain, Francielle Vargas, Ali Hürriyetoğlu, and Usman Naseem. 2024. Extended multimodal hate speech event detection during russia-ukraine crisis-shared task at case 2024. In *7th Workshop on Challenges and Applications of Automated Extraction of Socio-Political Events from Text, CASE 2024*, pages 221–228. Association for Computational Linguistics.

Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Surabhi Adhikari, Hariram Veeramani, Mehwish Nasim, and Usman Naseem. 2025. Large language models (llm) in computational social science: prospects, current state, and challenges. *Social Network Analysis and Mining*, 15(1):1–30.

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)*.

Thilini Wijesiriwardene, Hale Inan, Ugur Kursuncu, Manas Gaur, Valerie L. Shalin, Krishnaprasad Thirunarayan, Amit Sheth, and I. Budak Arpinar. 2020. Alone: A dataset for toxic behavior among adolescents on twitter. In *Social Informatics*, pages 427–439, Cham. Springer International Publishing.

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.