

# Identifying toxic behaviour in online games

Patrick Schrottenbacher

*schrottenbacher@em.uni-frankfurt.de*

*Goethe University, Frankfurt, Germany*

**Keywords:** Online video games · Toxicity · Graph neural networks

## **Abstract**

In online video games toxic interactions are very prevalent and often even considered an imperative part of gaming.

Most studies analyse the toxicity in video games by analysing the messages that are sent during a match, while only a few focus on other interactions. We focus specifically on the in-game events to try to identify toxic matches, by constructing a framework that takes a list of time-based events and projects them into a graph structure which we can then analyse with current methods in the field of graph representation learning.

Specifically we use a Graph Neural Network and Principal Neighbourhood Aggregation to analyse the graph structure to predict the toxicity of a match.

We also discuss the subjectivity behind the term toxicity and why the process of only analysing in-game messages with current state-of-the-art NLP methods isn't capable to infer if a match is perceived as toxic or not.

## Acknowledgments

Thanks to Sinika Decot, Reed Oken, Eli Ismailov and Len Martin for enduring my 3am rants.

Thanks to my friends and family in helping me annotate my dataset.

Thanks to the TTLab team for providing guidance.

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Related Work</b>	<b>5</b>
2.1	Toxicity . . . . .	5
2.2	Graph representation learning . . . . .	6
<b>3</b>	<b>Methodology</b>	<b>7</b>
3.1	Video game . . . . .	7
3.2	Graph construction . . . . .	8
3.3	Datasets . . . . .	9
3.4	Neural Network . . . . .	9
3.5	Annotation & Survey . . . . .	10
3.6	Training . . . . .	12
<b>4</b>	<b>Results &amp; Discussion</b>	<b>14</b>
4.1	Detoxify dataset . . . . .	14
4.2	Survey . . . . .	16
4.3	Annotation . . . . .	19
<b>5</b>	<b>Limitations</b>	<b>22</b>
<b>6</b>	<b>Future Work</b>	<b>22</b>
<b>7</b>	<b>Conclusion</b>	<b>23</b>
<b>8</b>	<b>Appendix: Glossary</b>	<b>27</b>
<b>9</b>	<b>Appendix: Node Vector Construction</b>	<b>27</b>
<b>10</b>	<b>Appendix: Survey</b>	<b>27</b>
<b>11</b>	<b>Appendix: Annotation</b>	<b>33</b>
<b>12</b>	<b>Appendix: Source Code</b>	<b>33</b>

# 1 Introduction

Nowadays video games are one of the most utilized forms of media. They are especially popular among the demographic of teens and younger adults[8][23]. While not necessarily being a new sub-genre, specifically online video games have recently been gaining more and more popularity.

Online video games allow a massive amount of people to interact with each other in real-time. This allows people to engage in cooperative or competitive play with their friends but also with strangers.

There are various ways in which people can interact with each other in video games. Besides interacting in the game world itself, video games often also allow for text-based and audio-based communication between players.

These can be used to chat or convey ones intent but are also often used to vent frustration and anger towards other players. This can lead to toxic interactions, especially in competitive games[18].

A lot of studies have focused on online video games precisely to study those interactions.

Some of these have focused on analysing video game chat logs, often classifying the various messages and actors as toxic or non-toxic. Some also take other forms of in-game interaction between players into account but mostly only those that serve the point of direct communication[18].

Other forms of interactions between players end up being completely discarded, or unaccounted for, or at the very least end up not being used to derive toxicity from them.

These are precisely the aspects that this thesis will seek to focus on.

Since players can choose to act toxically in a given video game match, these actions are often correlated with other events and the progression of the match itself[18]. Other players may perceive these kinds of matches as toxic based on the behaviour of the toxic actor. This brings us to our main hypothesis:

- H: Video game matches that contain toxic behaviours have a detectable, abstract internal structure. Be it either because certain chain of events, cause players to show toxic behaviours, or because toxic behaviours from players influence their own and or other player actions.

To test this hypothesis, we create a way to efficiently represent an entire match in graph form, taking inspiration from recent work in the field of temporal graph embeddings. Alongside this we prepared two main datasets labeling various matches according to their toxicity with different strategies. One of which is derived from current state-of-the-art sentiment analysis tools while the other will be derived from annotations provided by a group of individuals. We then analysed the dataset based on a survey that was provided with the aim of highlighting potential biases in the annotation process, as well as to gain further insight into the subjective perception of toxicity and how applicable current sentiment analysis tools are at performing the task at hand.

Following that we created and trained a neural network based on the datasets to predict whether a match is perceived as toxic purely on the events that take place in a given match.

The paper itself is organized as followed: In Section 2 we will review previous works. This is further split into two sub sections: In Section 2.1 we will define and highlight the importance of toxicity in online video games as well as the efforts to analyse toxicity in online spaces. In Section 2.2 we will focus on recent works to create and evaluate graph embeddings. Following this, we will provide a detailed overview of our methodology in Section 3, also shedding insight into our annotation process in Section 3.5.

Finally, we will show and discuss our findings in Section 4 while highlighting current limitations as well as future improvements in Section 5 and Section 6 respectively, providing our conclusion in Section 7.

## 2 Related Work

In this section we will first offer a definition and overview for the term *toxicity* alongside its various nuances. In the second part we will give a short insight into graph representation learning with a focus on graph neural networks.

### 2.1 Toxicity

Toxicity is a widespread term used to describe negative behaviours from players. To be more specific it often is supposed to be an umbrella term for aggressive, harassing and or insulting communication[25] while others more vaguely describe it as “spreading a bad mood”[11]. This vagueness around the concept of toxicity can be further attributed to at least three reasons[4]. First the range of anti-social attitudes being very broad and complex, second the game communities and their habits varying across the board and third players’ ability and frustration being a trigger for toxic behaviour.

Toxicity is a widespread issue in the gaming space across a multitude of games[25][12] and it can often transcend the space of a particular video game itself into other platforms[4]. There are multiple aspects that cause toxicity in online video games. One of which is the aspect of anonymity, which has been shown to be a major factor in both qualitative studies, with the focus on how it affects the community around a particular game[25], as well as quantitative studies[11]. Another supposed reason for toxic behaviours are stressful in-game experiences as well as frustration[12][10]. These are often experienced during the duration of a match and cause a transition from non-toxic to toxic behaviour[13]. In general, the amount of people that play video games is increasing[8] partially due to the success and popularity of video games that are competitive at their core[4][18]. While only a minority of a community tends to show toxic behaviours[28], it constitutes a major issue for video game companies and for players, due to the quality of games decreasing and subsequently, the enjoyment of the players themselves[10][11][25][4][14]. In particular, affected communities

of this tend to be minorities e.g women partially due to the established stereotype of the white, male gamer[1][23].

Toxicity in video games is so prevalent that there exists a certain normalization of its presence in video game spaces[25]. Often players treat it as an essential and imperative part of a game and/or gaming in general. While there have been a lot of studies focusing on online toxicity most focus on social networks[28] while the specific study of toxicity in video games in comparison is quite unexplored[10]. For video games specifically most papers focus on processing and analyzing in-game chat messages[17][26] with a few considering platform-wide consequences[4]. These studies have also shown that toxicity is linked to in-game success by additionally analysing match statistics and in-game events [18].

## 2.2 Graph representation learning

The field of graph representation learning is vast and contains many different subsets of studies. One of which is deriving embeddings for the nodes in a graph. Similar nodes, should in this case obtain a similar embedding. These embeddings can then be used for further machine learning tasks such as node classification or community detection. One of the most well-known frameworks for this is Node2Vec[6].

Temporal networks are a subset of networks. In these networks, every edge also has a corresponding timestamp attached to it. Each edge then represents an event between two nodes. These networks are often more effective at representing real systems. Weg2Vec[27] (weighted event graph 2 vector) provides a framework to derive an event embedding from a network. It does this by transforming a temporal network into a weighted event graph, with the weights being derived from the time difference as well as the co-occurrences of each event. Existing architectures can then be used on this graph to derive various embeddings. For instance, in the paper they sample a set of environments for each event and use these as input to a Skip-Gram model[16] to obtain event embeddings.

Graph Neural Networks (GNNs) are another field related to studying graph structures. These models aim to create a way to process graph structures with existing neural network methods[20] and have had a significant impact on graph representation learning. A framework that was derived from this is message passing neural networks[5]. In short, message passing neural networks work by looking at all neighbouring node embeddings (messages) for every node and then aggregating them with an aggregate function (e.g. sum). All pooled messages are then passed through an update function which typically is a neural network. One issue with this method is that some aggregators can fail to differentiate between the incoming messages. For instance, assuming we use a mean aggregator, it can fail to differentiate if two distinct nodes, each with two neighbours receive the messages  $(2, 2)$  and  $(0, 4)$  since the mean of both is 2.

An architecture that aims to resolve this issue, is Principal Neighbourhood Aggregation (PNA)[2]. This tries to resolve the issue by deriving a way to utilize

multiple aggregators together. In particular, PNA achieves this by introducing scalars, which are functions of the number of incoming messages for a node. These scalars are then multiplied with the value from the aggregator to amplify or attenuate incoming messages. Together this then results in a process that better understands graph structures as well as retain neighbourhood information.

## 3 Methodology

We will now present our methodology that we used to try to test our hypothesis. First we will introduce the video game which we chose to analyse delving into our process of deriving various events from its matches in Section 3.1. After that we will outline how we construct our graph in Section 3.2 followed by our method on deriving our datasets in Section 3.3 as well describing our choice of neural network to train on the datasets in Section 3.4. Lastly we will define our annotation process alongside its accompanying survey in Section 3.5

### 3.1 Video game

To test our hypothesis we analyzed matches from the video game *Team Fortress 2*<sup>1</sup>.

*Team Fortress 2* is a multiplayer first-person shooter game developed by *Valve* released in 2007. Since its inception users have created a community driven competitive game mode, which we will use as our basis.

We chose this game given the fact that we want to retrieve as much information as possible from a given match. *Team Fortress 2* is one of the only games for which nearly every competitive match is published on third-party websites such as *logs.tf*<sup>2</sup> or *demos.tf*<sup>3</sup>.

*Demos.tf* hosts entire recordings of matches which can be replayed in-game. *Logs.tf* on the other hand provides various statistics about a match. It derives these from log files which are created during the game.

The game itself is set up as followed:

There are two teams, the BLU team and the RED team. Each team tries to capture an objective. These objectives are different depending on the game mode.

Once an objective has been captured the respective team earns a score point. Whichever team, at the end, has the most points, wins.

During the game, players from different teams can damage and temporarily, eliminate each other, usually for a certain amount of seconds.

Each team also has a player that can heal their teammates with their *Medi Gun*. While healing, this player charges their *Medi Gun* which, when fully charged, allows them to grant their teammates invulnerability for a certain amount of

---

<sup>1</sup><https://www.teamfortress.com/>

<sup>2</sup><https://logs.tf/>

<sup>3</sup><https://demos.tf/>



time. All of these different interactions and events are then logged in a file which is then uploaded to the *logs.tf* service.

We used these files to extract the various events that occurred in a given match. The specific events that we used were:

1. A player eliminating (killing) another player
2. A player capturing an objective
3. A player using their *Medi Gun* charge
4. A player dying while having their *Medi Gun* charge
5. A player sending a message

These events in particular were chosen for their great relevancy and impact in a match.

### 3.2 Graph construction

Once our methodology on how to gather our events had been established we next focused on constructing our graphs.

We can view our list of events as a temporal network connecting either two players or a player and an action. Similar to the method presented in Weg2Vec we now try to project our temporal network into an event graph. To do this we represented each event as a node in our graph. The embeddings of our node are a 5-dimensional vector representing the type of event. For more details see Section 9. We then draw an edge if the events occurred in a certain time span  $\Delta t$ . More specifically we draw an edge as follows:

1. Test if the events are successive and within a time frame  $\Delta t$  of each other and share the same players
2. If they are and the initiator of the event is the same in the two events, label the edge with a '1'
3. If they are and the victim of the event is the same in the two events, label the edge with a '0'

Under these rules we create our graph for a given match. The aim of this is to represent the structure of a match in regard to both time and space.

We will later show the effects of different values for  $\Delta t$ .

Alongside this we also create a player graph. For this, we take each player that has played in the sampled matches and represent them as a node. We then draw a weighted edge between the players that played on the same team for each match. The weight at the end then represents the number of times these players have played with each other. We then used Node2Vec to derive a node embedding.

We did 10 random walks per node with a walk length of 20. Our nonnormalized

probabilities to return or move away from a source node were  $\frac{1}{0.7}$  and 1 respectively.

These embeddings, with a vector size of 32, were then used to enhance our edge features for our annotation dataset.

Since in our match graph we draw an edge whenever the same player is associated with two events we will extend the feature vector of that edge with the corresponding player’s node embedding. We also append a 1 to our edge vectors if the player played on the *Blue* team and a 2 if our player played on the *Red* team, reserving the value 0 as a fallback. We do this to study the effects, which a more feature-rich event graph might have on our predictions.

### 3.3 Datasets

Next we created the dataset. For this, we sampled matches from *logs.tf* with ids ranging from 3,000,000 to 3,400,000, which represent matches from the 15th of August 2021 up until the 20th of April 2023. All of the log files were then parsed with our parser and then subsequently categorized into 5 different languages. These languages included English, French, German, Spanish and Russian. For this we used *Lingua*[24].

We then used *Detoxify*[7] to analyse every chat message that was detected as English by *Lingua*. This allows us to develop a base understanding of the toxicity of the messages written in a given match. However it does not per se offer us any insight if the given video game match would be considered toxic. Since in our research we couldn’t find any papers classifying entire video game matches as toxic, we created our own scoring system based on the message classification from *Detoxify*.

First, we scored every message by the following criteria:

- 1 if the message was classified as containing *toxicity* with a certainty of at least 0.8
- 5 if the message was classified as containing *severe\_toxicity* with a certainty of at least 0.8
- 0 otherwise

All of the scores were then summed up and divided by the total number of messages leading to our final *match-toxicity-score*.

From this dataset we then sampled our main datasets which we will describe further in Section 3.6.

Since we also want to explore the accuracy of this scoring system to classify matches as toxic we also sampled 4 thousand matches from this dataset so they can be annotated by humans. We describe this entire process in Section 3.5.

### 3.4 Neural Network

After outlining a framework to create a graph representation of a given match as well as creating a dataset we then proceeded to construct a neural network

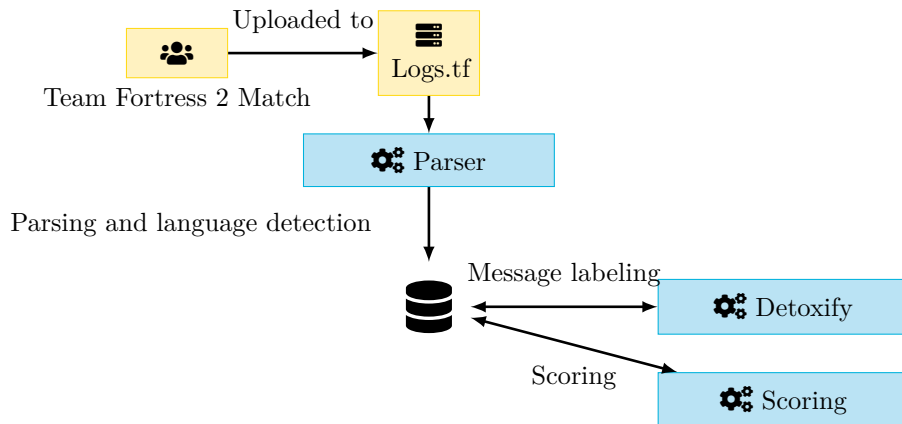


Figure 1: Illustration of the process to derive our dataset. The log files from a match first get uploaded to the *logs.tf* service. We then download and parse these log files additionally detecting the primary language before we store them in our database. Finally we label the messages with *Detoxify* and derive our overall score.

which will be trained to detect toxic matches in our dataset.

The neural network we created was quite simple. First, we had our 5 dimensional node input vectors, followed by two PNA Convolutional[2] layers. Each of their outputs was normalized by a *BatchNorm*[9] followed by a rectified linear unit (ReLU). We chose the PNAConv layers specifically since they are designed to understand the structures of graphs while also allowing us to take advantage of our edge features. Importantly they have also been proven to be effective for datasets with and without edge features. After our PNAConv layers, we placed 3 linear hidden layers. With the dimensions 5, 10 and  $n$  respectively, with  $n$  describing the dimension of our output layer. For the activator functions we always chose ReLu's.

For our binary classification  $n$  equaled 1, while for our multiclass classification problem  $n$  equaled 3. A sketch of the neural network can be seen in Figure 2. The aggregators we chose for our PNAConv layers were 'mean', 'min', 'max' and 'std' with the scalers 'identity', 'amplification' and 'attenuation'.

### 3.5 Annotation & Survey

For the annotation we first created a sub dataset from our previously established dataset. Since our method of scoring showed that, most likely, only a few matches might be considered toxic we sampled our annotation train-dataset with a bias to encourage a more even distribution between toxic and non-toxic matches.

Besides that, it also allows us to better compare our two datasets, once the annotation process has been completed. In total we sampled and aimed to

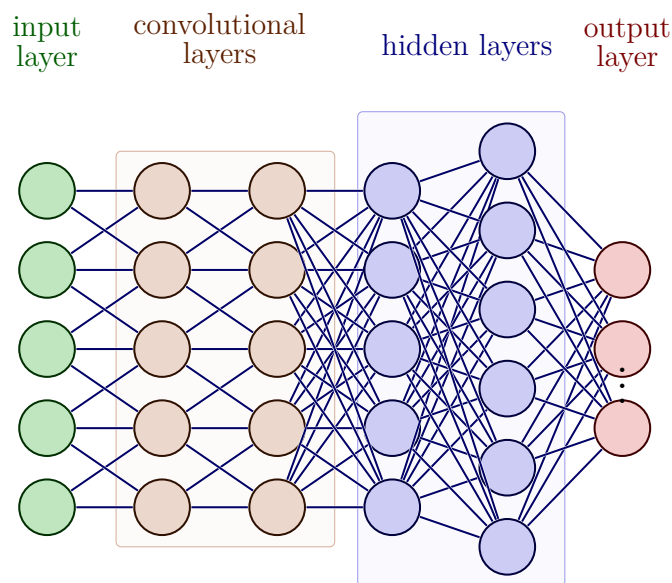


Figure 2: Sketch of the neural network we constructed.

annotate 4000 matches. Our training dataset included 2800 (70%) matches with 1200 matches (30%) used for the validation and test datasets. Our training dataset was split up as follows:

1. 70% of matches were sampled randomly
2. 15% of matches with a *match-toxicity-score* between 0.1 and 0.3
3. 15% of matches with a *match-toxicity-score* greater than 0.3

This aims for an even distribution while reducing the impact of potential biases introduced by sampling based on the scoring of *Detoxify*. We next let people annotate the dataset.

### Annotation

To allow various individuals to annotate the dataset we created a web page. The page first prompted people to enter a name and then allowed them to annotate various randomly sampled matches. The main challenge in the annotation laid in the presentation of the data at hand. Specifically the main trade-off was the amount of information presented versus the time it takes to annotate and comprehend the given information.

While an ideal annotation process, in terms of the information presented to the end user, might have involved showing them a recording of the entire match, this was not deemed feasible, given the amount of time it would take to annotate a match, in context with the time and resources available in the project's scope.

We hence chose to structure the annotation process as follows:

The annotation page consisted of two sections. The first one was a list of messages. Each could be annotated as *neutral*, *slightly toxic*, *toxic* and *extremely toxic*. A *Not Applicable* option (N/A) was also offered. Importantly the users also had the option to view the events that happened up to 20 seconds prior to a given message.

The second section asked the user to annotate the entire match with the same possible answers mentioned prior. In case there were issues with a given match that did not allow them to properly annotate a message and or the entire match they were also able to flag this being the case by clicking a checkbox. A sample image of this can be seen in Section 11.

The idea behind this was twofold. Firstly we wanted to create a dataset of annotated matches to better test our hypothesis while secondly also providing a way to compare the message annotations to the annotations from *Detoxify*. This should help us judge how accurate common sentiment analysis tools are at labeling video game chat messages.

## Survey

Alongside this annotation process there also was a mandatory survey participants had to fill in. This existed to clarify potential biases in the dataset as well as to expose the participant’s perception of toxicity in video games. Especially since as referenced in Section 2 the concept of toxicity is quite vague. The questions asked can be grouped into two sections: In the first section, we asked them questions about their playtime as well as which games and genres they typically engage in. This existed to get a baseline understanding of their habits and the types of video games they play. In the second section, we asked them about their personal perception in regards to online video games, competitive online video games and finally the games they themselves play.

More specifically we enquired if they perceive video games as places that tolerate toxic behaviour as well as places where toxic behaviour is manifested. Both of these questions existed to further investigate potential biases. The full survey can be seen in Section 10 We will discuss our results and findings specifically for our survey in Section 4.2.

## 3.6 Training

To finally evaluate our model we trained it on the following datasets:

First, we created two datasets based on the *match-toxicity-score* derived from *Detoxify*. One dataset contained 5000 samples while the other contained 10000 samples. On these datasets, we then trained both a binary and a multiclass classifier. For the binary classification, we used a threshold value of 0.3, to decide whether a match was toxic, according to our *match-toxicity-score*. For the multiclass classification problem, we chose the thresholds 0.3 to classify a match as *toxic* and 0.2 to classify a match as *slightly toxic*. Since our dataset is highly imbalanced we oversampled our training dataset. We also made use of

early stopping with a patience of 20, which was activated after 5 epochs. All our models were trained with the Adam optimizer and a binary or cross-entropy loss. The initial learning rate was  $10^{-4}$  and our batch size was 64 for our dataset with 5000 samples and 128 for our dataset with 10000 samples. Alongside this, we also reduced our learning rate by a factor of 0.1 if our validation loss hasn't improved, with a patience of 10. To evaluate these models we used the ROC-AUC score. For our binary classification task, we also used the PR-AUC score. We chose the *ROC-AUC* score specifically to get a better understanding of our model's general performance and the *PR-AUC* score, since it gives a better metric to understand a model's performance on an imbalanced dataset. For our multiclass classifier, we calculated the average *AUC* of all possible pairwise combinations of classes to ensure it's insensitive to our class imbalance. We trained the same neural network on our annotated dataset. In case we had two annotations for a match we took the floored average. We also only used the classes *neutral*, *slightly toxic* and *toxic* since we had too few cases of the label *extremely toxic*. In case a match was labeled as *extremely toxic* we classified it as a *toxic* match. We will use this dataset to compare our findings.

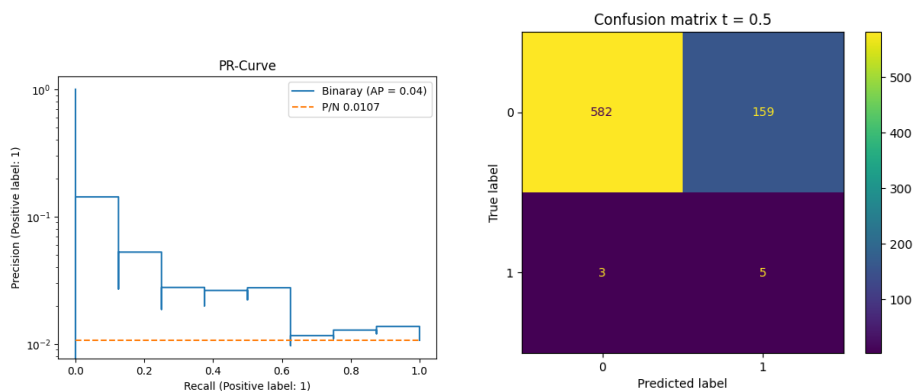
## 4 Results & Discussion

We will now present our results and discuss our findings. In the first section, Section 4.1 we will present our findings from our *Detoxify* datasets. We will then in Section 4.2 delve into our findings from our survey before finally showing and discussing our findings from the annotation data in Section 4.3

### 4.1 Detoxify dataset

First we'll look at the results of the binary classification from the Detoxify dataset. Since we, as previously mentioned, have a highly imbalanced dataset measuring and comparing our models based on their accuracy isn't ideal. Instead, we will mainly focus on the ROC-AUC and the PR-AUC score.

As we can see in the upper table in Table 1 our ROC scores all are quite acceptable yet our PR scores, which are the ones we're more interested in, are quite low. This is partially due to our low amount of actual toxic matches, that being around  $P/N = 1.07\%$  of matches, but also due to bigger issues with the accuracy of the classification.



(a) Plotted PR-Curve for our Detoxify dataset with  $\Delta t = 5$ .

Note the precision axis being in log scale. The low resolution is the result of our low amount of positive datapoints.

(b) Confusion matrix for our Detoxify dataset with  $\Delta t = 5$  and a probability threshold of 0.5.

Unsurprisingly we achieve a high amount of true negatives but our false positive rate is very high.

Figure 3

In Figure 3 we can see more precisely why our PR score is so low. In general, we achieve a very low precision and while it is better than our baseline across the board it isn't by much. In our confusion matrix with a probability threshold of 0.5 we can take a more educated guess as to what is happening: We achieve a decent true negative rate of 78.54% but the amount of false positives, is relatively high having a great impact on our precision.

One thing we can see in our Table 1 is that our PR scores don't seem to be effected strongly by a higher  $\Delta t$  threshold. Yet our ROC score tends to slightly increase. When we look at our multiclass scores we can see a similar trend

Type	Dataset	$\Delta t$	ROC-AUC	PR-AUC <sub>0.0107</sub>
Binary	Detoxify@5K	5	0.6468	0.0272
Binary	Detoxify@5K	10	0.6477	0.0181
Binary	Detoxify@5K	15	0.6755	0.0244
Multiclass	Detoxify@5K	5	0.4858	—
Multiclass	Detoxify@5K	10	0.5588	—
Multiclass	Detoxify@5K	15	0.5756	—
Type	Dataset	$\Delta t$	ROC-AUC	PR-AUC <sub>0.0200</sub>
Binary	Detoxify@10K	5	0.7090	0.0682
Binary	Detoxify@10K	10	0.7527	0.0833
Binary	Detoxify@10K	15	0.7449	0.0652
Multiclass	Detoxify@10K	5	0.6033	—
Multiclass	Detoxify@10K	10	0.6134	—
Multiclass	Detoxify@10K	15	0.5996	—

Table 1: Scores for the binary classification problem for our Detoxify dataset. The scores were calculated with sklearn[19] `roc_auc_score` and `precision_recall_curve` functions which calculate the scores for various probability thresholds. Note our very low  $P/N$  ratio of 0.0107.

emerging although a lot less pronounced, partially due to our lower scores in general. A lot of toxic remarks tend to be preceded by kill events[18] which could explain our findings. By increasing the timespan  $\Delta t$  we include more data that might be related to toxic behaviour hence our model is more effective at making accurate predictions.

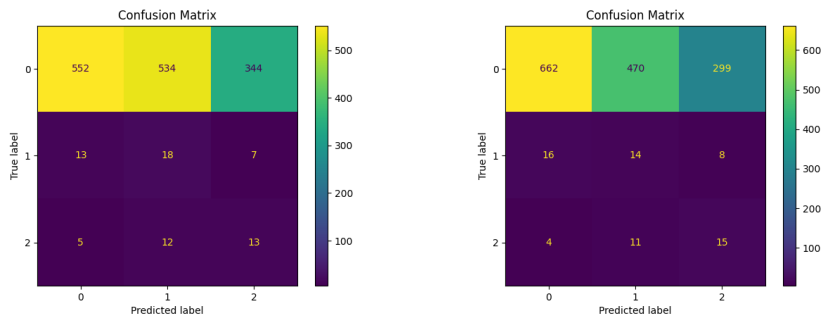
Another notable thing is that our scores tend to improve measurably when including more samples. This makes sense considering that with more matches in our dataset we also increase the diversity of matches; most importantly our toxic matches. This likely increases the chance our model can distinguish between the various classes.

If we look at our confusion matrices in Figure 4 we can get a clearer picture of how our model classifies our matches and where possible problems reside. Across the board we can see a similar pattern to our binary classification task emerging.

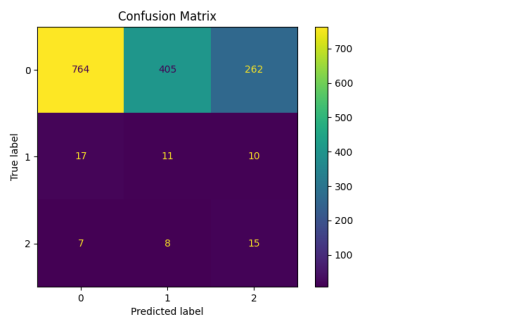
Although this time we can more clearly see that a lot of our neutral matches are also categorized as *slightly toxic* alongside *toxic*.

For our other cases we can see a slight tendency for our *toxic* matches to be labeled as *toxic* alongside *slightly toxic* while our *slightly toxic* matches tend to be labeled as *neutral* or *slightly toxic*. This does seem to hint at our model being able to understand and differentiate between toxic and non-toxic structures although it is difficult to concretely derive any such conclusions for certain. We can also see that our model starts to perform worse with a  $\Delta t$  of 15 suggesting





(a) Confusion matrix for our *Detoxify* dataset with 10000 samples and  $\Delta t = 5$ . (b) Confusion matrix for our *Detoxify* dataset with 10000 samples and  $\Delta t = 10$ .



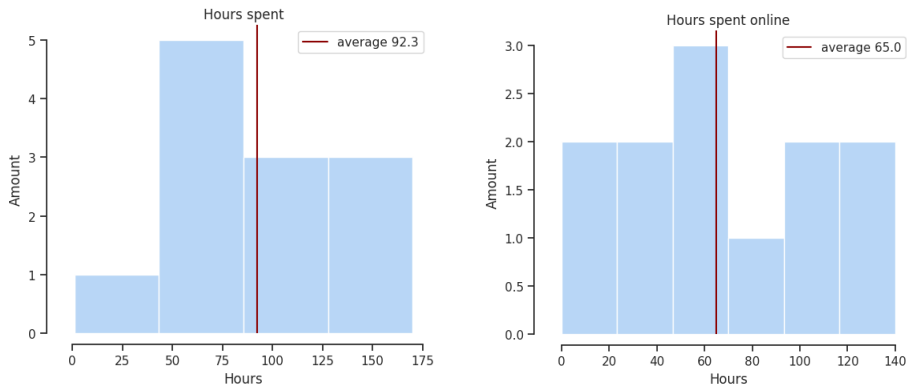
(c) Confusion matrix for our *Detoxify* dataset with 10000 samples and  $\Delta t = 15$ .

Figure 4: Confusion matrix for our *Detoxify* dataset with 10000 samples and different  $\Delta t$  values. The label 0 represents our *neutral* label, the label 1 our *slightly toxic* label and 2 our *toxic* label.

that we might, at some point, include too many unimportant data points. We will look at our survey and annotation results next to see the possible reasons as to why our model might be performing sub-optimally.

## 4.2 Survey

In total we had 14 participants who filled out our survey of which all 13 said they play video games and 12 that they play online video games. The first metric we want to look at is the number of hours our participants spend playing video games, on average, in a given month. As we can see in Figure 5 we have a very high average in the number of hours played per month, for both video games, as well as online video games. Therefore we should consider our annotations to be biased in that regard. Another notable thing is that the amount of hours spent playing online video games (see Figure 5b) is quite evenly distributed. This



(a) Reported amount of hours spent playing video games in a month. With 4 bins. The red line denotes the average amount of hours spent between all participants.

(b) Reported amount of hours spent playing **online** video games in a month. With 6 bins. The red line denotes the average amount of hours spent between all participants.

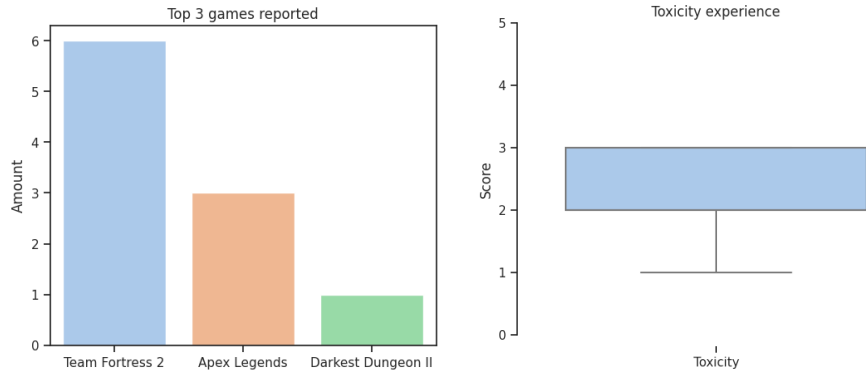
Figure 5: Reported amount of hours spent playing video games in a month.

should help eliminate some of the biases we might perceive due to our players spending different amounts of time playing video games.

Next we will look at the answers in regards to toxic behaviours in video games which can visually represented in Figure 7. Focusing on participants' opinion on whether toxic behaviour is manifested in video games, we can deduce that for competitive games, our participants very much agree with this being the case, while in the games they personally play they tend to disagree with that statement more (see Figure 7a). The answers in regards to if online games manifest toxic behaviour were inconclusive.

A similar pattern can be seen in regards to our participants' opinion on whether toxic behaviour is tolerated in video games which we can see depicted in Figure 7b. Although most notably our participants tend to all agree that toxic behaviour is tolerated in competitive and online games, we can yet again see that the results are more inconclusive for the games our participants personally play.

This indicates, together with our results in Figure 6, that our participants engage in games that they perceive as being less toxic compared to other games, therefore they might consciously avoid toxic games or perceive the games they themselves play as less toxic due to their exposure to them. When looking at the specific games our participants choose to play in Figure 6a we can see that a lot of them are familiar with the game we originate our dataset from. The genres are also in line with this and support a bias in regards to *FPS*, *Puzzle* and *Strategy* games being played.



(a) Top 3 most mentioned games the participants regularly play. (b) Participants own perception of toxicity in the games they play. A higher score correlates with a higher amount of toxicity being perceived. A score of 0 denotes no toxicity being perceived.

Figure 6

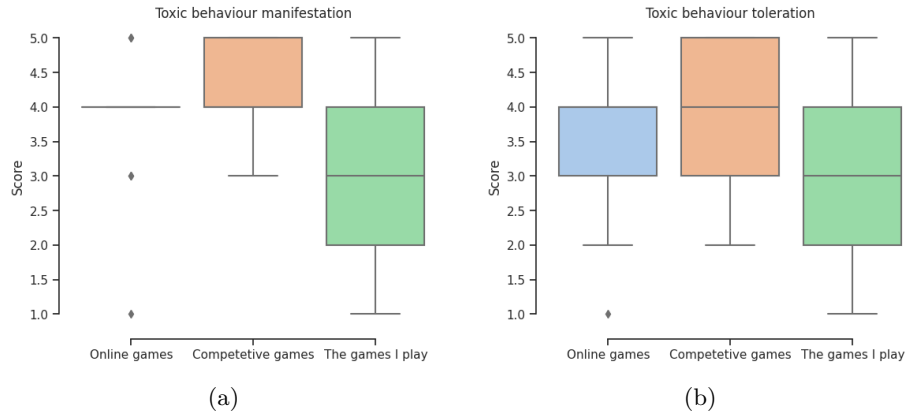


Figure 7: Results of the survey for the questions in regards to if games manifest or tolerate toxic behaviours. A higher score correlates with a higher level of agreement. A score of 3 denotes uncertainty.

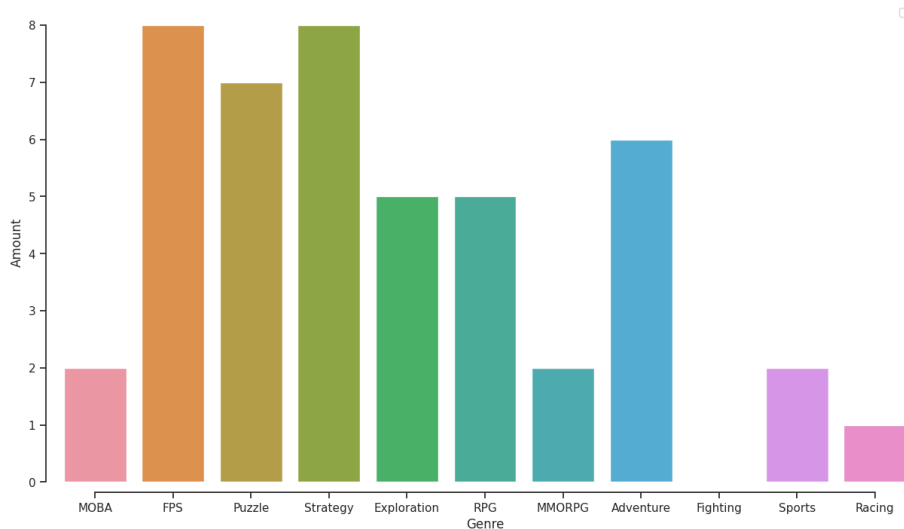


Figure 8: The genres of the video games participants often engage in.

### 4.3 Annotation

While we initially planned to annotate 4000 matches we only managed to annotate 687 individual matches, with 825 annotations in total. Around 17% of matches were annotated twice which will help us better understand the subjectivity behind annotations. One of the first evaluations we will take a look at is the correlation between our previously established *match-toxicity-score* and the actual annotations provided.

For this we plot the *match-toxicity-scores* in relation to the annotations done on the randomly sampled dataset. As we can see in Figure 9 our *match-toxicity-score* does seem to correlate with our annotations, in fact, we can derive a Pearson correlation coefficient[3] of 0.356, indicating a low positive correlation between our *match-toxicity-score* and our classes. Importantly though a lot of the thresholds between the various categories aren't clearly defined and overlap a lot. Even worse: a lot of the matches that were annotated as *neutral* have a *match-toxicity-score* way greater than our average for the *slightly toxic* category. This combined with our already low sample size of toxic matches in our *Detoxify* dataset might explain one aspect as to why our model wasn't able to provide accurate predictions, since some of the matches we labeled as toxic were in fact, according to our annotations, not toxic at all. Another aspect we wanted to consider in regard to our participants' behaviour was the correlation between the amount of hours a person plays video games and how likely they are to annotate a match as toxic. To test analyze this we scored each annotation from 0 to 4 (0 being *neutral* and 4 *extremely toxic*) for each participant that annotated more than 30 matches and calculated the average. We then

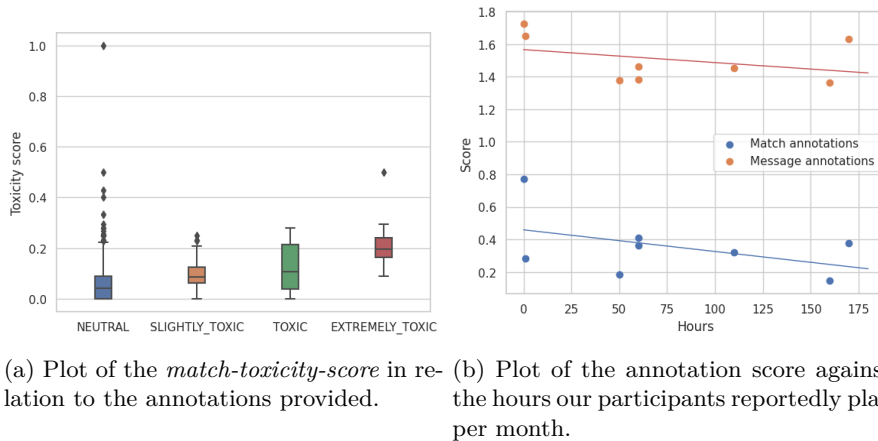


Figure 9

plotted these average scores against their hours played to derive the graph seen in Figure 9b. From this, we can then derive Pearson correlation coefficients and receive a value of  $-0.453$  for our match annotations and a value of  $-0.3649$  for our message annotations. This infers that the more hours someone spends playing video games the less likely they are to perceive a match or message as toxic.

Another aspect we want to look at is our disagreement between our annotators. In the randomly sampled matches we annotated about 126 matches twice and our annotators disagreed on the annotation of those about 27% of the time. Most of these disagreements existed for the labels *neutral* (about 24 times) and *slightly toxic* (about 32 times). This suggests that those two labels are not clearly defined, hence we might also struggle to classify those two labels correctly with our model.

Let’s now look at the predictive performance of our model with our annotated dataset. Similar to our previous dataset we’ll look at our ROC-AUC scores alongside the confusion matrix to judge the performance.

In Table 2 we can see the ROC scores we achieved. Notably, we can see that we achieve a higher ROC score than with our previous dataset. The confusion matrix can be seen in Figure 10a, in which we can now see that we can label neutral games a lot more accurately when compared to our previous dataset. It also seems to be able to better predict our *slightly toxic* matches, yet our *toxic* matches still can’t be predicted very accurately. It is however very difficult to make any real comparisons due to our low amount of samples.

Finally let’s see if we can improve our scores by supplying more data to our graph representation. For this, we enhance our graph with our player graph node embeddings as discussed earlier in our methodology. As we can see in Table 2 we do increase our ROC score marginally but most importantly we

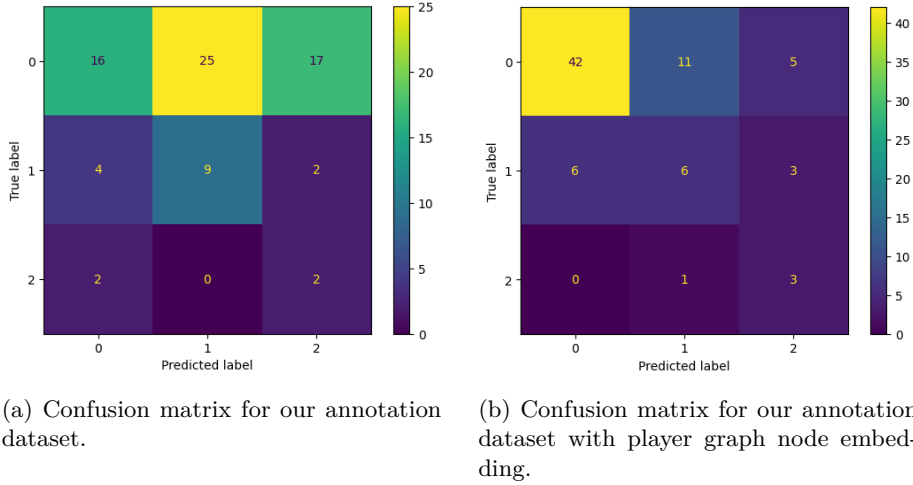


Figure 10: Confusion matrix for our annotation datasets. In this 0 represents our *neutral* label, 1 our *slightly toxic* label and 2 our *toxic* label.

reduce the number of false classifications for our neutral labels drastically (see Figure 10b).

Type	Dataset	$\Delta t$	ROC-AUC
Multiclass	Annotation	10	0.6957
Multiclass	Annotation-Enhanced	10	0.7237

Table 2: Scores for the multiclass classification problem for our annotated dataset.

This does seem to hint at the fact that our model can now better discern between the match structures by supplying more data to our graphs. This is also supported by the fact that our model is being trained for longer before we stop due to our validation loss not further declining. We trained our model for approximately 43 epochs on the normal annotation dataset compared to 71 epochs on our enhanced dataset.

In general, we can say that our model performs a lot better on our annotated dataset, although the short sample size makes it difficult to derive any concrete conclusion. It is notable though, that many of the wrongly classified matches tend to instead be classified as adjacent labels. For instance a *neutral* match being falsely classified as a *slightly toxic* match. Which further suggest that our model is properly recognizing our structures.

## 5 Limitations

A big limitation is how we currently decide to detect toxic behaviours. Currently, we only focus on the chat messages and the surrounding events to decide if a match is toxic or not. As pointed out in other papers there are other behaviours, sometimes dependent on the game, that are considered to be toxic in the community. Some of these are universal across a genre (e.g t-bagging and spawn camping) while others are more specific to a game (e.g the ability to taunt someone as an in-game action). Alongside this we've only been collecting data from one video game so the results of our study can only be generalized to a certain extent. Another limitation to consider is that we have people from different demographics annotating our matches. As shown in the results most of our annotators are familiar with video games but their exposure or amount of time spent playing games differ a lot. We could already show that there tends to be a decent amount of disagreement between the annotators especially when it comes to *neutral* and *slightly toxic* labels. This also highlights another limitation which is the subjectivity behind labeling matches as toxic, which will inherently make it difficult to properly annotate matches. The model we use to train our data on is also very simple and we did not partake in any hyperparameter optimisations or considered other model structures. We could almost certainly improve our model's predictive power with these methods. As well as that the amount of data we managed to annotate is also somewhat limited. As we could show with our *Detoxify* datasets our model accuracy increased considerably when increasing the amount of matches in our dataset. Another limitation specifically in regards to the results of our survey is that the survey was constructed without any expertise. While there existed an intent behind the questions asked the phrasing might not be optimal to gather the data we wanted to collect.

## 6 Future Work

As discussed in Section 5 one of our big limitations currently is the lack of taking other forms of toxicity into account besides the messages. In the future, we would like to detect these types of community and game-specific toxic events so we can include them in our graph representations. This should also help our annotators to more accurately judge if a match should be considered toxic or not.

Another big aspect we want to look into is how well this process works in other video games. Therefore we would like to apply our process to other games such as Dota 2 and League of Legends, both of which have been the subject of prior studies. This should grant us a better overview of the applicability of our system across various different video games.

It would also be interesting to generalize our system further to beyond the space of video games, to the space of general multi-user interactions to classify interactions that might be easily discernible to a human observer, but for which

it is difficult, laborious or impossible to define concrete metrics to automate the process. An example of this would be credit card fraud but it could also be applicable for more direct interactions between humans e.g. in simulation based learning, particularly in VR[15].

Since the amount of matches included in our annotation dataset was quite low we'd like to extend the size of our dataset as well, especially once we've added more types of toxic events to our graph representation. Alongside this, we would like to further optimise our machine learning model e.g with hyperparameter optimisations. Both should help to increase our model's predictive strength.

## 7 Conclusion

While we could show that our model can detect and discern between different structures in our matches, we can't definitively state that our hypothesis has been proven, partially due the low amount of samples. We could however show promising aspects such as our model being able to discern between toxic and non-toxic matches with reasonable accuracy.

At the same time we established a method to represent video game matches in a graph format while also outlining some aspects that can increase or decrease the predictive strength of our model to infer toxicity.

Alongside this we also outlined some of the potential issues in relation to the term toxicity and its perception. In particular we have shown a negative correlation between the amount of hours a participant plays video games and their likelihood to annotate a match as toxic, alongside the fact that toxic matches are not always clearly discernible from non-toxic matches. Finally we have also shown how current NLP methods aren't particularly well suited to declare video game matches as toxic and would advocate against the use of such methods for video game matches.

## References

- [1] Braithwaite, A.: 'seriously, get out': Feminists on the forums and the war(craft) on women. *New Media & Society* **16**(5), 703–718 (2014). <https://doi.org/10.1177/1461444813489503>, <https://doi.org/10.1177/1461444813489503>
- [2] Corso, G., Cavalleri, L., Beaini, D., Liò, P., Velickovic, P.: Principal neighbourhood aggregation for graph nets. *CoRR* **abs/2004.05718** (2020), <https://arxiv.org/abs/2004.05718>
- [3] Freedman, D., Pisani, R., Purves, R.: *Statistics (international student edition)*. Pisani, R. Purves, 4th edn. WW Norton & Company, New York (2007)



- [4] Gandolfi, E., Ferdig, R.E.: Sharing dark sides on game service platforms: Disruptive behaviors and toxicity in dota2 through a platform lens. *Convergence* **28**(2), 468–487 (2022). <https://doi.org/10.1177/13548565211028809>, <https://doi.org/10.1177/13548565211028809>
- [5] Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. *CoRR* **abs/1704.01212** (2017), <http://arxiv.org/abs/1704.01212>
- [6] Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. *CoRR* **abs/1607.00653** (2016), <http://arxiv.org/abs/1607.00653>
- [7] Hanu, L., Unitary team: Detoxify. Github (2020), <https://github.com/unitaryai/detoxify>
- [8] Homer, B.D., Hayward, E.O., Frye, J., Plass, J.L.: Gender and player characteristics in video game play of preadolescents. *Computers in Human Behavior* **28**(5), 1782–1789 (2012). <https://doi.org/https://doi.org/10.1016/j.chb.2012.04.018>, <https://www.sciencedirect.com/science/article/pii/S0747563212001227>
- [9] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR* **abs/1502.03167** (2015), <http://arxiv.org/abs/1502.03167>
- [10] Kordyaka, B., Jahn, K., Niehaves, B.: Towards a unified theory of toxic behavior in video games. *Internet Research* **30**(4), 1081–1102 (Jan 2020). <https://doi.org/10.1108/INTR-08-2019-0343>, <https://doi.org/10.1108/INTR-08-2019-0343>
- [11] Kordyaka, B., Kruse, B.: Curing toxicity – developing design principles to buffer toxic behaviour in massive multiplayer online games. *Safer Communities* **20**(3), 133–149 (Jan 2021). <https://doi.org/10.1108/SC-10-2020-0037>, <https://doi.org/10.1108/SC-10-2020-0037>
- [12] Kordyaka, B., Laato, S., Weber, S., Niehaves, B.: What constitutes victims of toxicity - identifying drivers of toxic victimhood in multiplayer online battle arena games. *Frontiers in Psychology* **14** (2023). <https://doi.org/10.3389/fpsyg.2023.1193172>, <https://www.frontiersin.org/articles/10.3389/fpsyg.2023.1193172>
- [13] Kwak, H., Blackburn, J.: Linguistic analysis of toxic behavior in an online video game. In: Aiello, L.M., McFarland, D. (eds.) *Social Informatics*. pp. 209–217. Springer International Publishing, Cham (2015)
- [14] Kwak, H., Blackburn, J., Han, S.: 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*. p. 3739–3748. CHI '15, Association for Computing Machinery, New

- York, NY, USA (2015). <https://doi.org/10.1145/2702123.2702529>, <https://doi-org.proxy.ub.uni-frankfurt.de/10.1145/2702123.2702529>
- [15] Mehler, A., Bagci, M., Henlein, A., Abrami, G., Spiekermann, C., Schrottenbacher, P., Konca, M., Lücking, A., Engel, J., Quintino, M., Schreiber, J., Saukel, K., Zlatkin-Troitschanskaia, O.: A multimodal data model for simulation-based learning with va.si.li-lab. In: Duffy, V.G. (ed.) Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management. pp. 539–565. Springer Nature Switzerland, Cham (2023)
- [16] Mikolov, T., Chen, K., Corrado, G., Dean, J., Dean, J.: Efficient estimation of word representations in vector space (2013), <https://arxiv.org/abs/1301.3781>
- [17] Murnion, S., Buchanan, W.J., Smales, A., Russell, G.: Machine learning and semantic analysis of in-game chat for cyberbullying. *Computers & Security* **76**, 197–213 (2018). <https://doi.org/https://doi.org/10.1016/j.cose.2018.02.016>, <https://www.sciencedirect.com/science/article/pii/S0167404818301597>
- [18] Märtens, M., Shen, S., Iosup, A., Kuipers, F.: Toxicity detection in multiplayer online games (12 2015). <https://doi.org/10.1109/NetGames.2015.7382991>
- [19] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011)
- [20] Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. *IEEE Transactions on Neural Networks* **20**(1), 61–80 (Jan 2009). <https://doi.org/10.1109/TNN.2008.2005605>
- [21] Schrottenbacher, Patrick, L.A.: Logs.tf Parser, <https://github.com/TheBv/logstf-parser>
- [22] Schrottenbacher, P.: Identifying toxic behaviour in online games, <https://github.com/thebv/toxic-video-games-gnn>
- [23] Souza, A., Pegorini, J., Yada, N., Costa, L., Souza, F.C.: Exploring toxic behavior in multiplayer online games: perceptions of different genders (10 2021)
- [24] Stahl, P.M.: The most accurate natural language detection library for python, suitable for long and short text alike. <https://github.com/pemistahl/lingua-py> (2023)
- [25] Thelander, J.: Gaming and toxicity in overwatch (2019), <https://urn.kb.se/resolve?urn=urn:nbn:se:hkr:diva-22153>

- [26] Thompson, J.J., Leung, B.H., Blair, M.R., Taboada, M.: Sentiment analysis of player chat messaging in the video game starcraft 2: Extending a lexicon-based model. *Knowledge-Based Systems* **137**, 149–162 (2017). <https://doi.org/https://doi.org/10.1016/j.knosys.2017.09.022>, <https://www.sciencedirect.com/science/article/pii/S0950705117304240>
- [27] Torricelli, M., Karsai, M., Gauvin, L.: weg2vec: Event embedding for temporal networks (2019)
- [28] Urbaniak, R., Tempska, P., Dowgiałło, M., Ptaszyński, M., Fortuna, M., Marcińczuk, M., Piesiewicz, J., Leliwa, G., Soliwoda, K., Dziublewska, I., Sulzhytskaya, N., Karnicka, A., Skrzek, P., Karbowska, P., Brochocki, M., Wroczyński, M.: Namespotting: Username toxicity and actual toxic behavior on reddit. *Computers in Human Behavior* **136**, 107371 (2022). <https://doi.org/https://doi.org/10.1016/j.chb.2022.107371>, <https://www.sciencedirect.com/science/article/pii/S0747563222001935>

## 8 Appendix: Glossary

- TP,TN,FP,FN: True Positive, True Negative, False Positive, False Negative
- Precision :  $\frac{TP}{TP+FP}$
- Recall/True Positive Rate (TPR):  $\frac{TP}{TP+FN}$
- False Postive Rate (FPR):  $\frac{FP}{FP+TN}$
- Specificity/True Negative Rate (TNR):  $\frac{TN}{TN+FP}$
- PR-Curve: Precision-Recall-Curve, Plot of the precision against the recall at various thresholds
- ROC: Receiver Operating Characteristic, Plot of the TPR against the FPR
- AUC: Area Under Curve
  
- Spawn camping: The act of immediately killing a player after they have spawned
- T-bagging: The act of crouching and 'un-crouching' on an opponents corpse

## 9 Appendix: Node Vector Construction

Our node vectors are constructed as follows:

$$\begin{bmatrix} 1, & \text{if the event is a } \textit{kill} \text{ event else } 0 \\ 1, & \text{if the event is a } \textit{charge used} \text{ event else } 0 \\ 1, & \text{if the event is a } \textit{medic died with charge} \text{ event else } 0 \\ 1, & \text{if the event is a } \textit{message} \text{ event else } 0 \\ 1, & \text{if the event is a } \textit{capture} \text{ event else } 0 \end{bmatrix}$$

## 10 Appendix: Survey

**1. Please specify a name.**

This must be the same name/alias you use when annotating later.

Next

2. Do you play **online video games**?

- Yes
- No

Do not wish to specify

[Back](#)

[Next](#)

3. Do you play video games?

- Yes
- No

Do not wish to specify

[Back](#)

[Next](#)

3. How much toxicity do you experience in the games you play?

None

Very Little

A decent amount

A lot

An extreme amount

No answer

[Back](#)

[Next](#)



Please take stance on the following statements:

**4. In \_\_\_ toxicity is tolerated.**

	Disagree	Slightly Disagree	Uncertain	Slightly Agree	Agree	No answer
Online video games	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Competitive online video games	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The games I play	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>


**5. \_\_\_ manifest toxic behaviors.**

	Disagree	Slightly Disagree	Uncertain	Slightly Agree	Agree	No answer
Online video games	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Competitive online video games	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The games I play	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>













[Back](#)

[Next](#)

## 11 Appendix: Annotation

TTLab Annotator Your annotations: 98 Total annotations: 540 

Please annotate the following messages:

#	Team	Name	Message	Annotation	Info
#4	Blue	Player 4	rude buddy	<span>Not Toxic</span> <span>Slightly Toxic</span> <span>Toxic</span> <span>Extremely Toxic</span> <span>N/A</span>	
#5	Blue	Player 4	i win	<span>Not Toxic</span> <span>Slightly Toxic</span> <span>Toxic</span> <span>Extremely Toxic</span> <span>N/A</span>	
#6	Red	Player 13	fair	<span>Not Toxic</span> <span>Slightly Toxic</span> <span>Toxic</span> <span>Extremely Toxic</span> <span>N/A</span>	
#7	Blue	Player 1	:D	<span>Not Toxic</span> <span>Slightly Toxic</span> <span>Toxic</span> <span>Extremely Toxic</span> <span>N/A</span>	
#8	Blue	Player 4	?	<span>Not Toxic</span> <span>Slightly Toxic</span> <span>Toxic</span> <span>Extremely Toxic</span> <span>N/A</span>	
#9	Red	Player 13	haaha	<span>Not Toxic</span> <span>Slightly Toxic</span> <span>Toxic</span> <span>Extremely Toxic</span> <span>N/A</span>	
#10	Red	Player 11	L	<span>Not Toxic</span> <span>Slightly Toxic</span> <span>Toxic</span> <span>Extremely Toxic</span> <span>N/A</span>	
#11	Red	Player 14	:D	<span>Not Toxic</span> <span>Slightly Toxic</span> <span>Toxic</span> <span>Extremely Toxic</span> <span>N/A</span>	
#12	Red	Player 3	gg	<span>Not Toxic</span> <span>Slightly Toxic</span> <span>Toxic</span> <span>Extremely Toxic</span> <span>N/A</span>	
#13	Blue	Player 4	gg	<span>Not Toxic</span> <span>Slightly Toxic</span> <span>Toxic</span> <span>Extremely Toxic</span> <span>N/A</span>	
#14	Blue	Player 5	gg	<span>Not Toxic</span> <span>Slightly Toxic</span> <span>Toxic</span> <span>Extremely Toxic</span> <span>N/A</span>	
#15	Blue	Player 8	gg	<span>Not Toxic</span> <span>Slightly Toxic</span> <span>Toxic</span> <span>Extremely Toxic</span> <span>N/A</span>	

How would you label the entire match?

Not Toxic Slightly Toxic Toxic Extremely Toxic N/A

Send annotation  I can't annotate this match

## 12 Appendix: Source Code

The source code and data is publicly available and can be found here: <https://github.com/TheBv/toxic-video-games-gnn>[22]

The source code for the parser can be found here: <https://github.com/TheBv/logstf-parser>[21]



Publiziert unter der Creative Commons-Lizenz Namensnennung - Nicht kommerziell - Weitergabe unter gleichen Bedingungen (CC BY-NC-SA) 4.0 International.

Published under a Creative Commons Attribution-NonCommercial-ShareAlike (CC BY-NC-SA) 4.0 International License.

<https://creativecommons.org/licenses/by-nc-sa/4.0/>