Benchmarking AI Techniques for Toxicity Detection in Online Games

Matteo Diana

Project Supervisor: Dr. Julian Frommel

Second Examiner: Dr. Dong Nguyen

Submitted in partial fulfillment of the requirements for the degree of

Master of Science in Game and Media Technology

Utrecht University
Student ID: 8255083

Utrecht University

Contents

Glossary

polysemy Capacity for a word or sentence to have different meanings based on the context. E.g.: going to to the *bar* / the *bar* is so low. 11

Tribunal Platform created by Riot Games for League of Legends. While it was active, users could read chat logs from players who have been reported multiple times and decide whether the player's behavior was toxic or not. This was one of the very few platforms where players could not only see what happened behind the scenes of a report system, but also have direct influence on it by voting. This system was ultimately disabled in early 2014. 13, 14

Acronyms

Away From Keyboard (AFK) Refers to players leaving the game, either for connection issues or as a trolling behavior. 17

CCSoft Okey Player Abuse (COPA) Database consisting of all the reports made by players in the online version of the tile-based game Okey.. 13, 14, 16

Command Line Interface (CLI) User interface where the interaction with the program or device happens via text, opposed to a menu with visual features (Graphical User Interface). 28

Fair Play Alliance (FPA) The Fair Play Alliance is a coalition of developers and gaming companies that are actively working to fight toxicity and create healthy environments for players. 8, 10

Human-Computer Interaction (HCI) Branch of science that studies how humans interact with technology. 7

League Of Legends (LOL) MOBA developed by Riot Games. 9, 10, 14

Multiplayer Online Battle Arena (MOBA) Type of game where two teams compete against each other in an arena. Often involves destroying enemy resources in order to obtain upgrades. 14

Natural Language Processing (NLP) Machine Learning techniques that are used to understand and manipulate human language.. 7

Acknowledgments

This thesis, which marks the conclusion of a long journey, owes its existence to the exceptional guidance of my supervisor, Dr. Julian Frommel, who advised me through the project with incredible passion, understanding, and energy. Working under his mentorship has been an absolute pleasure, and I am truly grateful for his invaluable contributions.
My sincere thanks to Dr. Dong Nguyen for agreeing to evaluate this thesis and for her helpful feedback.
I would like to express my appreciation to the Toxicity group for engaging in productive conversations during our weekly meetings. Special thanks to Michel Wijkstra for his help during the past months. I would also like to mention my peers Ilham and Nikola.
Additionally, thanks to Bertie Vidgen for generously providing me with a Key for the Rewire API.

Thanks to my family, especially my parents Nicola and Rossella. Your encouragement means the world to me.
Thanks to my girlfriend Elena for her love and support, and for always making me laugh with her fantastic Super Mario impression.
To Alice, who played a pivotal role in my survival during the initial months in the Netherlands. To the 228, for being a subpar accommodation but a beautiful home.
To all the great people that I have met during my years in the Netherlands.
To my friends back in Italy, daje Roma.
Finally, I believe that my laptop deserves some praise for surviving up to this point.

Abstract

Toxicity is a serious issue that affects millions of people worldwide. This issue is exacerbated in gaming environments, because of the competitive nature of online games and the normalization of negative behaviors. Detection of toxicity then becomes extremely important to effectively moderate game environments. Current techniques employed by gaming companies are not available to the general public. In academia, there is little standardization in terms of benchmarking. It is extremely important to compare different techniques to see how effective they are in a gaming context. Hence, the first part of this project will focus on building and comparing different models on the same dataset. CONDA, a dataset of DOTA 2 chatlogs, was chosen as the dataset of choice. For testing, I chose models that were commonly used for detecting toxicity in online spaces(Detoxify, Rewire, Perspective). These models were then compared to a finetuned instance of BERT. The different scores were computed and used to generate a new dataset containing all the scores for 8974 sentences. It was seen that BERT outperformed the other models. Afterward, I analyzed the strengths and shortcomings of each model, showing that the toxicity models could not generalize well to videogame slang.
Finally, I focused on multiclass classification. I compared two models of BERT that had varying levels of complexity to see how that impacted performance. Finally, zero-shot learning was performed using different instances of transformers.

Chapter 1

Introduction

Videogames are more popular than they have ever been, and their influence on society keeps growing stronger with time. Games are not only mere forms of entertainment: they improve psychomotor skills [22], can be used for therapy [10], and are currently employed for training workers, who can learn the ins and outs of a new job without worrying about making mistakes [2]. In the same way, games can be used to have fun with friends and, in the case of online games, build meaningful connections with people from all over the world [37].
However, people who are used to playing online games know that these communities are not safe spaces: it is extremely easy to encounter toxicity while enjoying a game. Toxicity is an umbrella term that describes antisocial behavior and harassment [34] [1]. It is not only limited to verbal aggression: it also includes abuse of the game mechanics, intentional feeding, and cheating. A large majority of players has been a victim of toxicity [16]. In particular, minorities are disproportionately affected [14] [36].
Toxic behavior is so pervasive that at this point it is almost normalized, with players becoming desensitized to it [4]. An indicator of the normalization of toxicity could be seen in the observation that the most experienced and skilled players are more likely to be toxic [41].
Given the detrimental effect of toxicity on players' enjoyment and even mental health, being able to identify it is crucial. Since some online games have millions of users playing at the same time, employing automated techniques for the detection of toxicity can be useful to make systems more scalable and trustworthy. Some games employ fully automatic detection systems, commonly keyword-based: for example, if a bad word is detected, the message might not be sent or the player might get muted for the rest of the game. However, this is just a stopgap, as these games usually also have to utilize other solutions in addition to it. The most commonly employed techniques involve a *reactive* report system, where action is only taken after a player gets reported. However, players tend to distrust the flagging system and often do not use it optimally [23].
Hate detection systems have been of great interest to the scientific community for a long time now. A review of these techniques was published in 2017 by Schmidt and Wiegand [40]. Recently, scholars have started focusing on identifying the inherent weaknesses of detection models. Content which is hard to detect, such as comments that employ code words or sarcasm, are grouped under the umbrella of veiled or covert toxicity [19] [27]. Sometimes, the definition also includes sentences containing typos or confusing word boundaries [18]. Research in this field has helped making models more robust and trustworthy.

In gaming literature, despite the large impact of toxicity on players' enjoyment and well-being, research regarding its detection is often overlooked. Moreover, there is little to no communication with industry stakeholders. There are no common benchmarks used

to understand how effective a model is. Minimal research has been done on the specific weaknesses of the models, and what users do to circumvent automatic detection while playing. Hence, I believe that the contributions of this thesis, listed below, could be useful to the Human-Computer Interaction (HCI) community. First of all, I will perform a review of the most commonly employed detection techniques in games. Afterward, an evaluation of the different zero-shot toxicity models on a single real-world dataset. We define as zero-shot toxicity models some language models (such as Perspective [46]) that are trained to detect toxicity, but have not been finetuned on the dataset at hand, because the wrights of the model can't be updated by users. These evaluation will be compared with models that were finetuned for the task at hand. Moreover, I will focus on the analysis of the weaknesses and strengths of each model. For example, this could help shed a light on the difference between toxicity in online games and in other online contexts.

1.1 Structure

Section 2 will focus on the literature review: the main focus will be on toxicity in games and what has been done to predict it. Furthermore, I will showcase what Natural Language Processing (NLP) techniques have been applied to predict toxicity in similar contexts and how they could be applied to the field of online games.
Section 3 focuses on research goals and hypotheses.
Section 4 gives insight into the structure of the CONDA dataset, which will be used for comparing the models. Section 5 delineates the methods section: the architecture of the models, features of the dataset, and course of action.
Section 6 highlights the results obtained.
Finally, Section 7 will showcase the discussion section. Section 8 contains the conclusion.

Chapter 2

Related Work

2.1 Toxicity

If one has experience with multiplayer videogames, it is easy to intuitively understand what is meant by toxicity. This umbrella term is used to describe a variety of disruptive behaviors, that range from insulting other players to cheating.

However, there is no clear consensus on the definition of toxic behavior.
Suler [43] first introduced the concept of **toxic disinhibition**, which defines antisocial behavior, including "rude language, harsh criticism, anger, hatred, even threats". This concept is introduced as the negative half of the **online disinhibition effect** (ODE), which explains how people feel less restrained and freer to express themselves on the internet. A good definition of toxicity is provided by Neto et al. [34]. In their work toxic behavior is defined as a behavior that happens when players are exposed to an event that generates frustration and anger, resulting in a harmful type of communication between peers.
Other words are used to express similar concepts. For example, **griefing** [17] refers to play styles that intentionally or unintentionally disrupt other players' experiences.
Cook et al. [13] use the word "trolling", and explain that the concept can be divided into verbal and behavioral trolling. Behavioral trolling refers to a category of non-verbal actions that deliberately put your team at a disadvantage, such as leaving the game or intentionally giving your opponents important resources. The authors highlight that, while verbal trolling is common all over the internet, behavioral trolling is somewhat unique to the context of games.
In their framework, the Fair Play Alliance (FPA) [1] states that, since the term "toxicity" is extremely colloquial and open to interpretation, they prefer to use the term **disruptive behavior**. Disruptive behavior, similarly to griefing, refers to conduct that is detrimental to the experience of players or the well-being of an online community.
Other papers use terms such as "anti-social behavior" [24], "cyberbullying" [32], "dark participation" [25], or "deviant behavior" [11]. Nonetheless, toxicity is still a word that is widely used in scientific literature. Furthermore, despite using different names, these definitions are quite similar. For this reason, throughout the thesis, these terms will be used interchangeably.
Other authors focus their research on the detection of **hate speech**. Based on the definition of hate speech (abuse or threat against a particular group), it could be said that it is a particularly harmful subset of toxicity.

After having provided a clearer definition of the terminology, it is time to explain why toxicity is such a core issue for players and gaming companies alike. The first, obvious reason, is that almost every player has experienced toxicity during their playtime. A recent study by the Anti Defamation League (ADL) [26] shows that 83% of the interviewed

subjects have experienced harassment in online games, and 71% reported *abuse and severe harassment*.
Being constantly subjected to negative behavior is extremely detrimental to players' well-being. The survey shows that a large percentage of players quit playing certain games, or outright avoid them, because of how toxic the community is (or is perceived). Moreover, it is reported that almost 20% of the subjects had felt isolated or alone after being exposed to toxicity, while 14% of them reported having suicidal thoughts as a consequence.
These statistics are in line with the findings of previous research. For example, Ortega et al. [35] highlight that cyberbullying experiences have serious consequences on the victims, often resulting in low self-esteem, worse performance in school, and, in worse cases, depression.

Furthermore, toxic behavior disproportionately affects minorities. In 2012, Consalvo [12] published an article that highlighted the concerning amount of misogyny that was happening in online spaces. She published a timeline of targeted attacks towards women and asked other scholars to keep track of other similar attacks, to have an archive that can be used to understand recurring patterns or behaviors. She elaborated that gamers are prone to behave in a toxic way towards women for two reasons. The first one concerns sexist beliefs about the proper place for women – what the author ironically defines as an *encroachment* of women and girls into what previously was a male-gendered space. The second reason focuses on the fears of players regarding how the gaming industry would change with the increasing presence of women.
Multiple authors have investigated how minorities cope with the constant harassment they are subjected to [14] [36] [33] [31]. The consensus is that players belonging to a minority group adopt strategies such as hiding their identity and avoiding verbal communication with other players to mitigate online harassment. Furthermore, the authors state that, after a while, players get desensitized to this negativity. Ortiz [36] states that this desensitization is reminiscent of "emotionally detached masculine coping strategies". A large majority of the players being interviewed added that this coping strategy was not their initial response to hate, but rather it was suggested by friends or family.

The ubiquity and seriousness of these issues make it crucial to educate players on how to behave responsibly. To do that, it is necessary to moderate the game environment. Good moderation is crucial to protect the well-being of players from the negativity that inevitably occurs when playing online games.

2.2 Moderation

Efficient content moderation is something that every company should take into consideration when developing a game. More recently, however, instead of focusing on moderation, companies have shifted their efforts to Player Dynamics. The change of paradigm is clear: instead of focusing on banning players who behave against the rules, gaming companies want to create a game that fosters positive interaction by default. As Weszt Hart, head of the Player Dynamics department at Riot Games, mentions in his article on design [20], it is easier to work on the gaming environment than to mitigate disruptive behavior once the game is released. This discourse perfectly makes sense when talking about developing new games. However, most players still keep playing old games, or new games that heavily take inspiration from old ones (such as reboots or remastered versions of classic games). For example, League Of Legends (LOL), which came out more than 10 years ago, keeps

growing in popularity, with 150 million active players across all servers last month.

For this kind of game, where the player dynamics were not prioritized during the early stages of the development process, the only way to limit the level of toxicity is by actively stopping it in its tracks. Even for new games, though, moderation is still needed, as it is (almost) impossible to make a game perfectly healthy by design.
On that regard, the FPA published a document that contains useful guidelines on how to moderate User-Generated Content [28]. The document focuses on **reactive moderation**, which refers to the kind of moderation that is prompted by users, who must always be able to notify the moderators that something against the guidelines happened during their game experience. This "notification" is commonly known as reporting or flagging.

The contents of the document show that content moderation should be highly scalable, as the quality of the report system should stay the same as the audience of a game grows larger. Furthermore, it shows that a larger audience could result in the human moderation team being overwhelmed. The obvious solution to these issues is automating the reporting pipeline or at least part of it. A majority of the most popular online games already employ automated models in some parts of the reporting pipeline. However, the document also states that it is vital that the flagging system must provide for action to be taken **reliably and consistently**.
A study conducted on LOL [23] players shows that most players do not believe this is the case. Players tend to distrust the report system, for a variety of reasons that can mostly be attributed to its black-box nature. Players do not know if their report was meaningful, what happened to the player that they reported, and report that they would rather prefer to report their experience to a human moderator.
Given the lack of information coming from videogame companies, it is not known for sure whether the report system is effective or not. Still, this research stems from the belief that, as observed by Kou and colleagues[23], players are not happy with how the report system in common multiplayer games is currently working. Hence, I believe it is important to develop tools that allow for accurate detection of toxicity. Employing these models could be a working mechanism to reduce toxicity.
Furthermore, I believe that the immediate and intuitive rating system employed by large language models, such as Perspective, could be used to detect toxicity in a more explainable way to users. If every message can be evaluated independently, it is easy to immediately tell which actions were ban-worthy, making the automatic evaluation more transparent to users.

2.3 Detection

Detection and moderation are two sides of the same coin. In order for content to be moderated, it needs to be understood.
Automated detection of negative behavior using Natural Language Processing (NLP) is a widely studied field. In 2017, Schmidt and Wiegand [40] published a survey on the automated detection of hate speech. They categorized the efforts of prior research into 8 different categories, based on the features used. These categories will be summarized below.

Simple Surface Features refers to papers using simple bag-of-words features, such as n-grams. N-grams are combinations of N contiguous words. For example, "f* you" is a toxic 2-gram. "hello world" is a non-toxic 2-gram. N-grams can be analyzed on either

word-level or character-level, where it was shown that character-level n-grams are more resistant to spelling variations.
Word Generalization is a technique where word clustering (such as Brown Clustering [8] or Latent Dirichlet Allocation [7]) is used to classify words. The newly created classes are then used as additional features in combination with n-grams. This technique is used to mitigate data sparsity issues that might happen when trying to detect toxicity on small pieces of text.
Sentiment analysis is often used as an additional classification, as it is closely related to hate speech. Most papers include a saliency analysis of the individual words (i.e. positive, neutral, negative) as additional features of their analysis.
Lexical resources is a technique where researchers add as an additional feature a binary flag that checks whether a negative word is part of the sentence. It is especially useful if the detection model wants to focus on a specific type of discrimination (i.e. based on ethnicity or sexuality).
Linguistic features involve some domain knowledge. These features are based on the usage of tokens or words that help give meaning to individual words. Alternatively, this category involves studying the relationships between non-consecutive words. In the context of hate speech, examples include relationships of the kind {*HateTarget, Stereotype*}.
Knowledge-based features are features that are needed to discern whether a message is harmful or not, based on the context. An example that could apply to the context of games, would be a sentence involving "killing" or "shooting" someone, which could be classified as hateful if taken out of context.
Meta-information refers to information that goes beyond the text of the post. It refers, for example, to information about the user who is writing the message, or the number of reactions to the post.
Multimodal information, in the paper, refers to analyzing the audio/video content that is attached to a text. In the gaming context, however, it could refer to analyzing the score of the game or the events that precede a message being sent.

Some of these labels, if shifted to the context of games, have different meanings. In particular, the labels "multimodal information" and "meta-information" are powerful predicting features, and are used in almost every paper, sometimes even without text information. Multimodal information, in this case, would refer to the game events, whereas meta-information would indicate the number of reports a player has received, or whether two players are friends or not. With that being said, these pieces of information are hard to obtain for researchers, with the current state of collaboration between industry and universities.

Detecting harmful content is a tough challenge, and many open problems still need to be solved. Each of the classes discussed by Schmidt and Wiegand can solve different issues, that were briefly introduced and can intuitively be understood. Vidgen et al. formalized them and published a list of open challenges in the context of hate speech detection [47]. The authors show that, when it comes to detection, the task is made especially hard by the subtleties of human communication and the shortcomings of models. Sarcasm or polysemy are issues of the first type, whereas weakness to spelling variations and dependencies between non-consecutive words are related to model weaknesses.

Lately, researchers have focused their efforts on trying to deceive the models to find out critical flaws. This type of research is extremely important in this context. For ex-

ample, in 2017, immediately after Google Perspective was released, Hosseini et al. [21] tested the system in an *adversarial setting*. The researchers actively tried to beat Perspective's detection system by using simple deception techniques. Their research showed critical flaws in Perspective's scoring system. Through some examples, they demonstrated that by simply changing a letter in a word, scores for otherwise identical sentences would change drastically. Replicating the same experiments now does not give the same results, which means that these weaknesses were identified and addressed.
Similarly, Gröndahl et al. [18] performed different kinds of adversarial attacks on seven state-of-the-art models trained on several datasets. Their findings show that the positive effect of employing complex architectures is not significant when compared to the importance of the dataset. Furthermore, employing data augmentation helps the performance of the model.

Given how important it is to evaluate the weak points of a detection model, several authors have published test suites that address this issue. For example, HATECHECK [39] is a test suite specifically built for hate speech detection models. The authors divided the 29 functionalities of their test suite into the following classes:

- Derogation
- Threatening Language
- Slur
- Profanity
- Negation
- Phrasing
- Non-hate group identity
- Counter Speech
- Abuse against non-protected targets
- Spelling variations.

When building a new model, this test suite can be used by stakeholders to understand how robust the system is against each of these issues.

2.4 Detection in games

As mentioned in theù previous sections, efficient and consistent detection of toxicity is crucial for online communities, including game environments.
For this reason, conducting research that allows to generate publicly available solutions to this problem is extremely important. Software such as Perspective are currently important resources, since they allow developers and small companies to implement some sort of automated content protection with a simple API call. Similarly, developing methods of toxicity detection that are tailor-made for games could be really beneficial to the community at large.
Where does the need to develop domain-specific solutions stem from?
When it comes to moderating text, it can be argued that the language commonly used by players during their experience is different to the one that is used in other contexts. That can be attributed to a variety of reasons, the first being linguistic. This refers to the usage of game-specific neologisms, such as the names of characters, strategies, or items. Another one is due to players often typing while playing, meaning that they have to be

quick in order to not lose too much time. This often results in typos or imperfect writing. These factors contribute in making game chats hard to classify correctly, calling for specialized resources.

Hereby I highlight the need for publicly accessible data, which would help the scientific community in generating high-quality models or frameworks for toxicity detection.
How is it possible to obtain this kind of data? When considering games like League of Legends, Overwatch, Valorant, and many others, the game chat effectively "disappears" after the game ends. Game replays have no chat function, and it is not possible to fetch chat messages using the API (only a few games allow it). This combination of issues has resulted in an almost total lack of large **game-specific** chat datasets available on the internet. Furthermore, there is also a complete lack of knowledge regarding **models** used by large gaming companies to detect toxicity.

Nonetheless, plenty of researchers have approached this problem, finding new and original ways to obtain data for their projects. These examples will be shown in the upcoming section. However, it should be noted most of the datasets that were used by the authors are not available. There are several reasons for this.
Some authors used resources that used to be accessible to everyone but are no longer online. These datasets include the Tribunal and the CCSoft Okey Player Abuse (COPA). Other researchers asked the videogame company to provide them data, under the agreement of not distributing them online. This allowed them to conduct "high-quality" studies, but these are not replicable, as the conversations are not open-sourced. Even though it would have been possible to ask these authors to share the dataset, it would still not be usable as a benchmark, since the data is not public domain. An example of this is the work of Canossa et al. [9].
Finally, some authors generated their own dataset by scraping hundreds of games and uploading their results on their personal websites. However, for reasons ranging from the websites not being maintained to the authors changing affiliations, these results are often not available.

Tables 2.1 and 2.2 show an overview of previous literature regarding the automatic detection of toxicity in games. Table 2.1 summarizes the papers that focus on detecting toxicity using **chat data**. Table 2.2 refers to papers that detect toxicity using behavioral data or other features.

Game datasets employing chat data

Blackburn and Kwak Studies by Blackburn and Kwak [6], as well as Stoop and colleagues [42], predicted toxic behavior using data from the Tribunal.
Blackburn and Kwak were among the first to predict toxicity in games. They employed Random Forests to predict the crowdsourced decisions of the Tribunal, using as input the features extracted from the tribunal, chat logs, and report logs. The results were satisfying, resulting in a detection of punish/pardon of 79.9% AUC.

Stoop et al. Stoop et al.[42] developed a framework (Harassment Recognizer, or HaRe) that keeps track of toxicity scores for each player over time. They modified the chat

Article	Game	Size	Dataset	Feature Set	Method	Target	Ground Truth	Criteria	S - W Category
Blackburn et al. [6] 2014	LOL	11M reports	Tribunal + /	In - game Stats Chat (U) User reports	Random Forest	Toxicity of player	Tribunal Evaluations	ROC , AUC	Sentiment Analysis
Märtens et al. [44] 2015	DOTA	10305 matches 7042112 words	Dotalicious + /	In - game Stats Chat (T)	n - gram analysis	Toxicity of team Impact on match	Annotation Match data	Accuracy	Surface Features
Balci and Salah[3] 2017	Okey	800000 matches	COPA /	Chat In-game stats	Bayes Point Machine	Toxicity of player	Annotation	Precision Sensitivity Specificity	Multimodal information
Thompson et al.[44] 2017	Starcraft	5046 (Annotated) 10742(Test)	Player games + /	Chat logs (U + T) Lexicon	Lexicon - based Sentiment Extractor	Toxicity	Annotation	Accuracy	Sentiment Analysis .
Murnion et al. [32] 2019	WOT	126091 5000 (annotated)	Wargaming Wotreplays + /	In - game Stats Chat (U)	SQL analysis Azure Sentiment Analysis	Toxic keywords	-	-	Lexical Resources
Stoop et al. [42] 2019	LOL	5000 matches	Tribunal + /	Chat logs (U) Context	RNN	Toxicity over time	Annotation	Precision Recall F - score	Linguistic Features
Weld et al. [48] 2021	DOTA 2	47000 utterances	CONDA	Chat (U + T) Context	NLU Models	Toxicity of utterance Label of sentence	Annotation	NLU - specific metrics	Word Generalization

Table 2.1: A review of toxicity detection tasks that focus on game datasets using chat data (and other features).
S-W classification indicates which one of the categories described by Schmidt and Wiegand [40] best describes the approach of the model being described.
+ indicates that the dataset was scraped by the creators of the article.
/ indicates that the dataset is restricted to the general public or no longer available.
U indicates that the chat logs are analyzed on the Utterance level.
T indicates that the chat logs are tokenized (added descriptive labels based on some dictionary).

dataset in order to evaluate consecutive messages from a single player as a single utterance. As their model of choice, they used the best-performing model from a Kaggle challenge for the classification of toxic comments. They then retrained it on their dataset of choice and evaluated the toxicity over time for different values of their threshold. The approach that they used for their network – using existing networks that were successful in other toxicity detection tasks, and then retraining it – is promising and will be utilized in this study as well.
However, neither study is fully replicable, because the dataset does not exist anymore. Furthermore, the Tribunal dataset was heavily skewed: only players who have been extremely toxic on multiple occasions ended up in the Tribunal, which means that these players were "outliers", with toxicity levels that are higher than what can be seen in average LOL games.

Märtens et al. A similar paper, which also focuses on Multiplayer Online Battle Arena (MOBA), was published by Märtens et al. [30]. They employed a dataset scraped from the website Dotalicious. The website no longer exists, and the dataset is not available anymore. They tokenized the game chat, annotating offensive n-grams[1] and checking for their presence in the game chat. Afterward, they used their results to analyze how the presence of toxicity affects the results of the game. Their method of checking for toxicity is simple and consistent, as it is token-based. This technique was used to quickly gather results for the main component of their study, which was an analysis of the correlation between toxicity and games. It is too simple to be employed in a real game. The paper also employs a letterset technique, where words are divided into sets of letters. For example, "idiot" uses the set of letters {I, D, O, T }. "iddiiiooooott" uses the same set of letters. This

[1]Given a list of "offensive" words, the authors only considered N-grams containing at least 1 element of the list. In this case, "f* you" would be a toxic 2-gram, whereas "f* this" would be a non-toxic 2-gram.

Article	Game	Dataset	Feature Set	Method	Target	Ground Truth	Criteria
Shen et al. 2020 [41]	World of Tanks	Wargaming + /	In-game Stats User reports Player rankings	Statistical analysis	Rank of toxic players Behavioral traits	-	-
Canossa et al. 2021 [9]	For Honor	Ubisoft /	In-game stats Movement data AFK time Chat* (No content)	RF	Toxicity of player Sanctions	Game data	Precision
Reid et al. 2022 [38]	Overwatch	Twitch + /	Audio data game data	RF Logistic Regression SVM	Toxicity (binary)	Annotation	accuracy recall F-score

Table 2.2: A review of all the toxicity detection tasks that focus on game datasets without employing chat data. These papers focus on behavioral data or other features.
+ indicates that the dataset was scraped by the creators of the article.
/ indicates that the dataset is restricted to the general public or no longer available.

technique is extremely similar to what Schmidt and Wiegand classify as character based n-gram, and their results show that this technique is useful in a game-specific context.

Thompson et al. Similarly, Thompson et al. [44] employ a lexicon-based technique to detect toxicity. They developed a simple and portable model that, instead of machine learning techniques, uses the sentiment score of each word to analyze the emotional salience (overall positivity or negativity) of a given sentence.
The authors focused on the game Starcraft 2 and used user-submitted replays as their dataset. Similarly to what was seen above, this dataset is not available to the general public.
The main advantage of a lexicon-based model is that it is extremely versatile, as it can be applied to any game with the proper modifications. The model uses a custom-made dictionary that contains game-specific words, which means that by modifying this file it is possible to change its scope. The authors mention that lowering the rate of false positives is a priority, as wrongly detecting something as toxicity is more harmful than not detecting something. By increasing the weight of the "false alarm" rate by a large margin, the model obtained an accuracy of around 80% in detecting the sentiment of sentences (on par with the accuracy of other detection systems, as seen above). However, this model needs a lot of trial-and-error with the choice of rules and cutoff parameters, along with constant updates to the game-specific dictionary as the game jargon evolves. Furthermore, increasing the cutoff margin by a large amount, as they did in this paper, means that the model only detects blatant cases of extreme toxicity.
Moreover, lexicon-based models require a lot of hand labelling, and can become unreliable in contexts where the jargon evolves quickly. For these reasons, in a real world scenario these models do not seem like feasible alternatives to machine-learning based ones.

Murnion et al. Further research involving semantic analysis of videogame data was proposed by Murnion et al. [32]. Their main focus was on generating a new dataset by continuously scraping chat conversations and in-game stats from World of Tanks websites. Afterward, they analyzed the dataset using SQL queries and sentiment text analysis services. For this second part of their analysis, they employed Twinword Sentiment Analysis and Text Analytics from Microsoft Azure Cognitive Services. However, the data they

scraped is currently not available, as the application the authors created appears to be broken [2].

Balci and Salah Balci and Salah [3] worked on the game Okey, a tile game similar to Rummikub. This game could be played online and had a chat option. They used the COPA Database, which contains the game logs of games where users have reported abusive behavior. The database consists of a variety of features: in addition to game chat, it is also possible to see the ratings of players, their profile information, and their match history. These features were used to determine whether the submitted complaints were genuine. The database is currently not available online.

Weld et al. More recently, Weld et al. [48] also helped the community by publishing a new dataset. They released CONDA, an annotated dataset subset of a larger DOTA2 dataset available on Kaggle[3]. The authors generated this dataset as follows: starting from the large corpus of conversations available on Kaggle, they discarded conversations that contained languages other than English. Then, they annotated it and applied optimization techniques similar to others that were mentioned above. For example, they combined consecutive messages from a single player into an individual utterance, in the same fashion as Stoop et al. [42]. They classify each utterance both on the sentence level and at the token level. The authors created six slot labels for their tokenization system: Toxicity (T), Character(C), Dota-specific (D), Slang (S), Pronoun (P), and Other (O). Then, they performed slot labelling for each word. Their game-specific dictionary was sourced by Märtens et al. [30]. For example, the sentence *"Can you help us report invoker ;)"*, after tokenization becomes "can (O), you (P), help (S), us (P), report (S), invoker (C), ;) (O)". There are 4 categories for an utterance: Explicit, Implicit, Action, and Other. Explicit refers to toxicity which is easily detected. Implicit refers to covert toxicity. Action and Other indicate normal game behavior or text that does not belong to the other categories and is not harmful.
The CONDA dataset is extremely useful, as it is one of the few annotated videogame datasets available on Github [4], and will be used as a baseline for this work.

The author tested the dataset by applying the paradigms of Natural Language Understanding (NLU). They try to detect **intent** and **slot filling**. In a way, this approach is a combination of the approaches seen above, as it is a simultaneous chat-based and lexicon-based analysis. The authors' approach, however, does not focus on the misclassifications of the different models. The objective of this thesis, on the contrary, is to compare techniques in order to understand the weaknesses of the different models.
More detailed information on the dataset can be seen in the dedicated section.

Game datasets without chat data

Other papers, mentioned in Table 2.2, use different approaches to detect toxicity. They are worth mentioning because of their originality – using types of data other than text – and the effort they made to collect meaningful data for their studies.

[2]Link here: https://asecuritysite.com/gamedata
[3]Link here: https://www.kaggle.com/datasets/romovpa/gosuai-dota-2-game-chats
[4]Link here: https://github.com/usydnlp/CONDA

Shen et al. Shen et al. [41] published a statistical analysis of data obtained from the creators of World of Tanks. The authors had access to the behavioral data of the games (e.g., type of battle, length, rank of players, and weapons) and to the timestamps concerning player reports. They focused on the correlation between toxicity and rank, discovering that more skilled players are more prone to being toxic. Furthermore, their study confirmed that players, as they are constantly exposed to toxicity, internalize these behaviors and start considering them a normal part of online games, becoming toxic themselves.

Canossa et al. Canossa et al. [9] published an article in collaboration with Ubisoft, which provided the dataset. The authors asked experts to select critical behavioral features and also performed statistical analysis to choose other factors that can be used to distinguish between sanctioned and unsanctioned players. Analyzing these features, they could distinguish between these two categories of players with, respectively, 85% and 91% accuracy. The features they used included activity models, match performance, Away From Keyboard (AFK) time, and movement speed. They also analyzed chat actions, but they only considered the number of messages per minute rather than their meaning.

Reid et al. Reid et al. [38] used **audio data** scraped from Twitch. The audio data were transcribed and then analyzed as text data, using Random Forest, Logistic Regression, and SVMs as their classification models. Their study was also successful, resulting in an 80% accuracy.

Chapter 3

CONDA

This chapter will focus on describing the ins and outs of CONDA, the dataset that will be used as the basis of this research. CONDA is a DOTA 2 dataset that was introduced by Weld et al. [48]. The dataset was scraped from Kaggle and is an annotated subset of English-speaking comments. The authors focused on both word-level tokenization and utterance-level classification. Initially, they performed a test run, where a subset of sentences was annotated by 4 player annotators and 2 non-players. The test showed that the agreement between player annotators only (Fleiss' kappa = 0.785) was higher than the one considering all 6 annotators (Fleiss' kappa = 0.755). This preliminary experiment highlighted the need of having some expertise of game-specific slang: for this reason, the annotation of the CONDA dataset was performed by annotators who had experience with games.

The authors opted for six slot labels for their tokenization system: Toxicity (T), Character(C), Dota-specific (D), Slang (S), Pronoun (P), and Other (O). Then, they performed slot labeling for each word. Their game-specific dictionary was sourced by Märtens et al. [30].

There are 4 categories for an utterance: Explicit, Implicit, Action, and Other. The labels of the dataset are generated by exclusion: the authors started by defining the Explicit Class as sentences that are outright offensive. Then, they said that the sentences in the Implicit class were offensive sentences that did not belong in the Explicit category. Action sentences are defined as sentences that are not I or E and contain action verbs. Other is defined as sentences that are not A, I, or E. In order simplify the class system, Explicit and Implicit can become a "toxic" label, whereas Action and Other can be transformed into the "not toxic" label. Utterances from the Implicit category are expected to be misclassified more than the other classes by the models. It is important to observe that this model only uses ALL chat, while the game also has team-only chat. However, DOTA2's API only allows obtaining data from the public chat. This is a setback, but not a major one, since it can be observed that there are a lot of messages containing toxicity.

Hereby I will conduct a preliminary analysis of the dataset, showing the class distribution, 1-grams, and 2-grams. The dataset contains 47000 sentences, divided in a 60-20-20 split for train-validation-test. However, the labels for the test dataset are not publicly available, as the authors used them for a competition on toxic language detection. For this reason, this section will be excluded from the dataset, focusing on the train and validation set. This is not a problem, as validation and test have the same amount of sentences and the APIs do not need finetuning. In other words, we will use the 26000 sentences from the training set to finetune BERT, and will test all the models on the 8974 sentences in the validation set. Table 3.1 shows the distribution of the labels for the different sets.

To give an idea of the distribution of words in the different sets, 1-grams and 2-grams are extracted and counted based on their occurrence within the dataset, providing insights

Class	**Total (n = 35895)**	**Train (n = 26921)**	**Validation (n = 8974)**
O	26611(74.14%)	19982 (74.22%)	6629 (73.87%)
E	4711 (13.12%)	3528 (13.11%)	1183 (13.18%)
A	2299 (6.40%)	1719 (6.39%)	580 (6.46%)
I	2274 (6.34%)	1692 (6.29%)	582 (6.46%)

Table 3.1: Intent class distribution. Total refers to Train + Validation.

into the language patterns specific to each class.

Figure 3.1 shows a list of the 15 most frequent 1-grams for each class. The list is unfiltered, with the exception of the separator "[SEPA]" being removed. This list contains a lot of very generic and non-informative words. These words, also known as stopwords, are commonly filtered in this type of analysis. For this reason, I performed filtering for each class using NLTK's collection of stopwords for the English language. [5]. The filtered version can be seen in Figure 3.2. This plot provides a lot of information on the characteristics of each class.

For what concerns the non-toxic classes, the 'Other' class contains a high number of utterances that are "formalities" of the game genre, such as saying "gg" or "wp" at the end of a match. The 'Action' class sees a high frequency of the words "report" and "commend", which are used to instigate players against a specific foe or to reward a player who distinguished himself.

In the toxic categories, it is clearly visible that the 'Implicit' category has an abundance of sentences containing the word 'easy', or 'ez' in any of its variations. Since the word is so predominant that it is hard to understand anything else from this class, additional analysis was performed on this dataset excluding the words 'easy' and 'ez'. Results can be seen in figure 3.3.

The 'Explicit' class is more varied, but almost all the words in the list are profanities of some kind.

Finally, Figure 3.4 shows the list of the most frequent 2-grams for said classes.

It should be remembered that the authors of the dataset joined together subsequent sentences from the same user. To keep track of this, they added the separator "[SEPA]" in the middle of sentences. This token will be removed when working with the pretrained models since these models expect to work with natural language - this is especially important because the separator was not part of the training dataset of these models. However, the separator was not removed when working with BERT, as these models can use this feature effectively. The reason for this difference is that all words are turned into embeddings during the preprocessing phase. Moreover, this token is present both in the training set and in the validation set.
Nonetheless, I expect this will not influence results, since this token does not provide any interesting features regarding the presence - or absence - of toxicity.

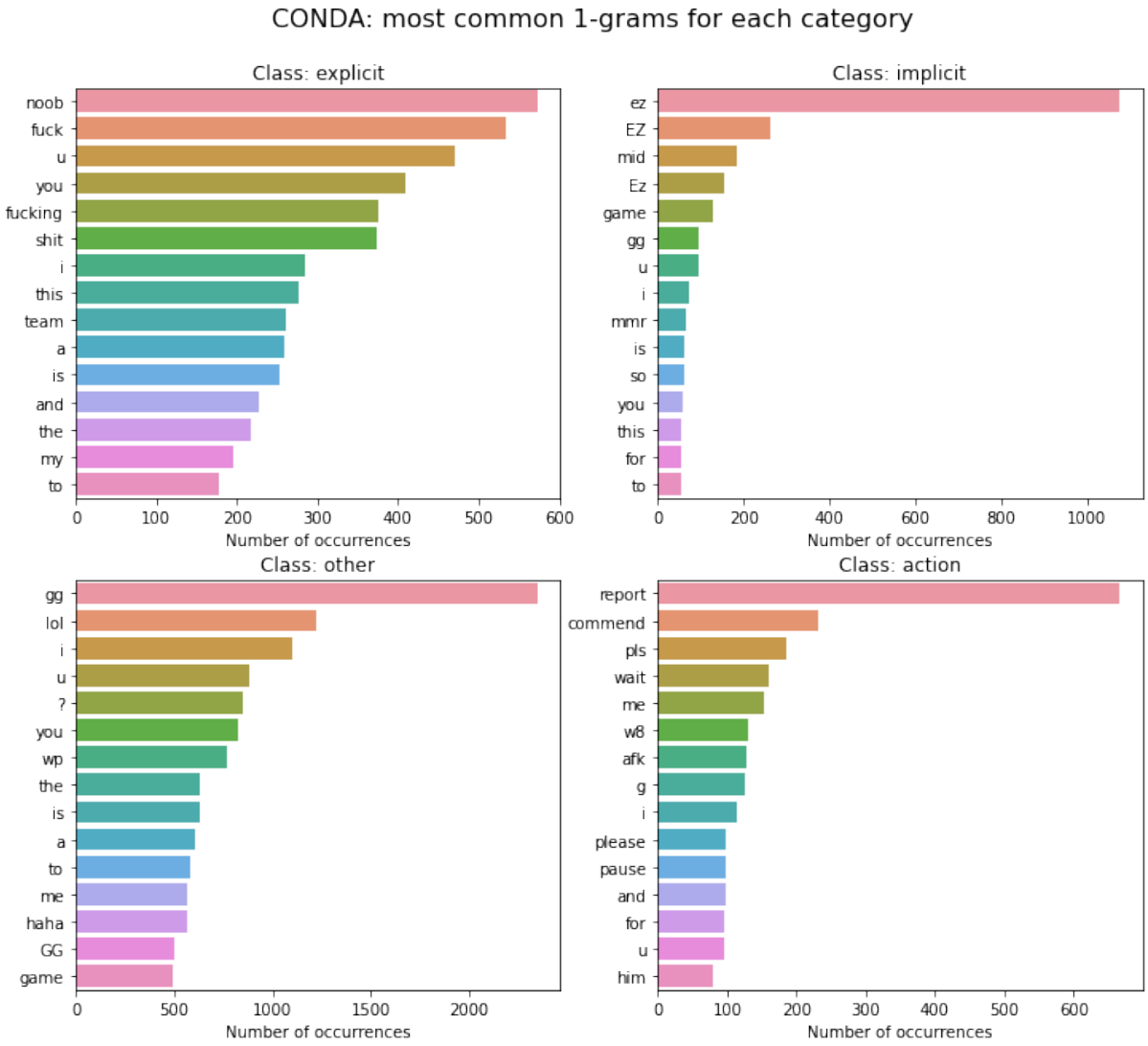


Figure 3.1: Top 15 Most Common 1-grams per Class on the CONDA Dataset. Unfiltered list, except for the separator "[SEPA]"

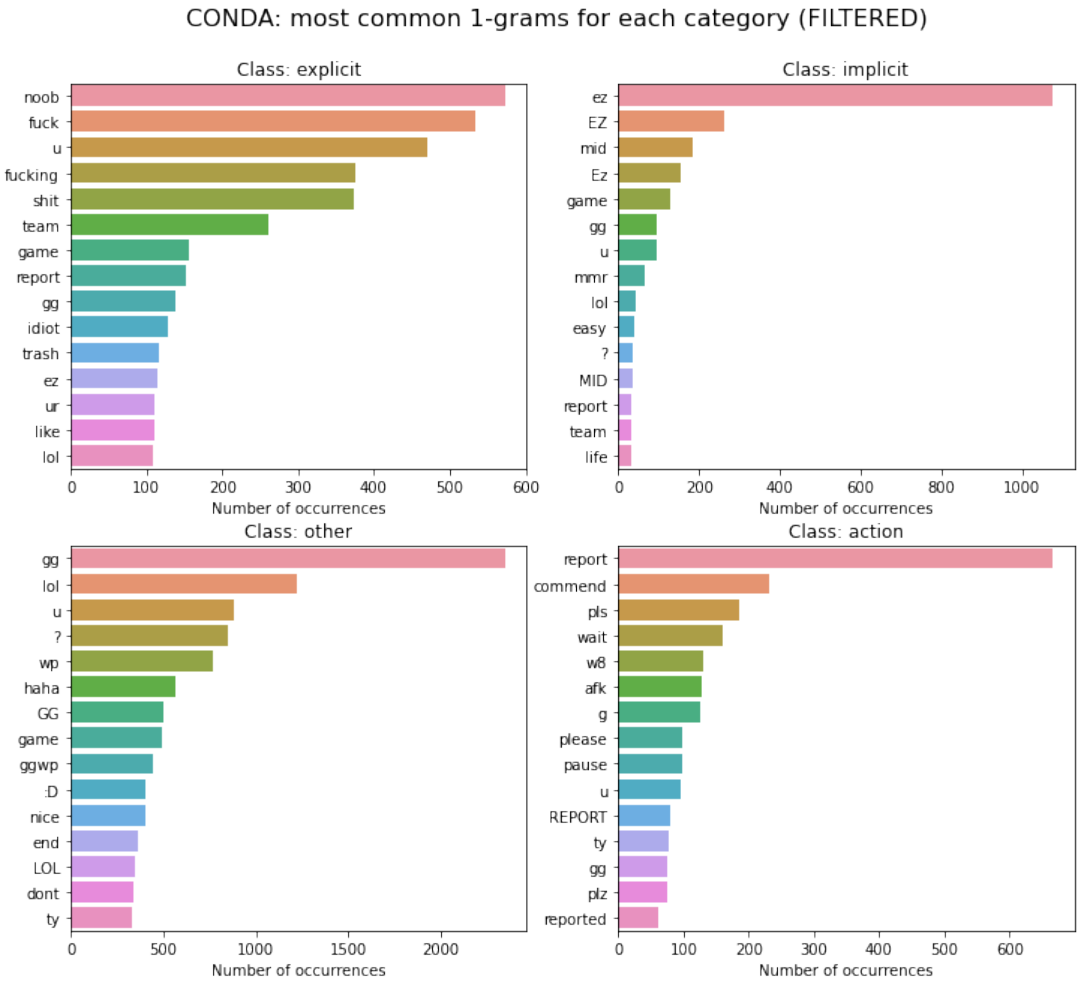


Figure 3.2: Top 15 Most Common 1-grams per Class on the CONDA Dataset. Filtered list, with stopwords being removed.

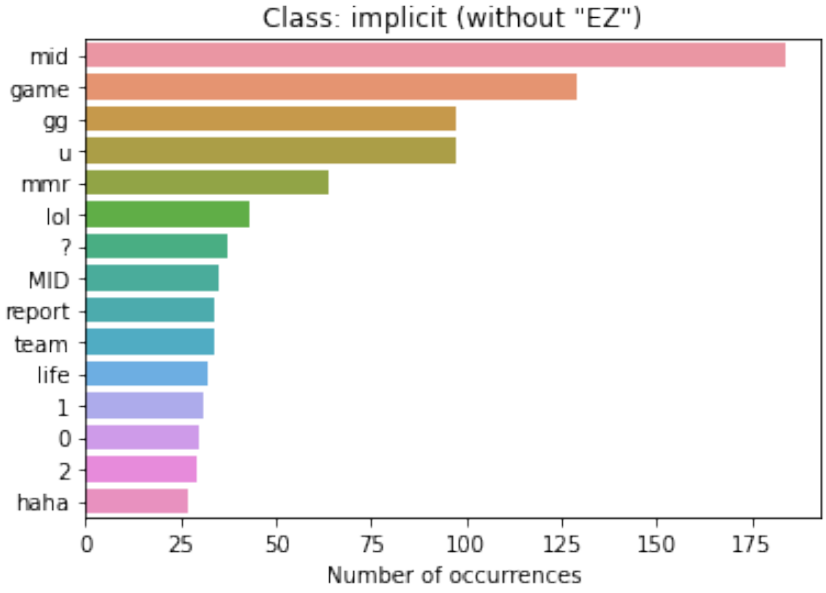


Figure 3.3: Most frequent 1-grams for the Implicit class, filtering all instances of the word "Ez".

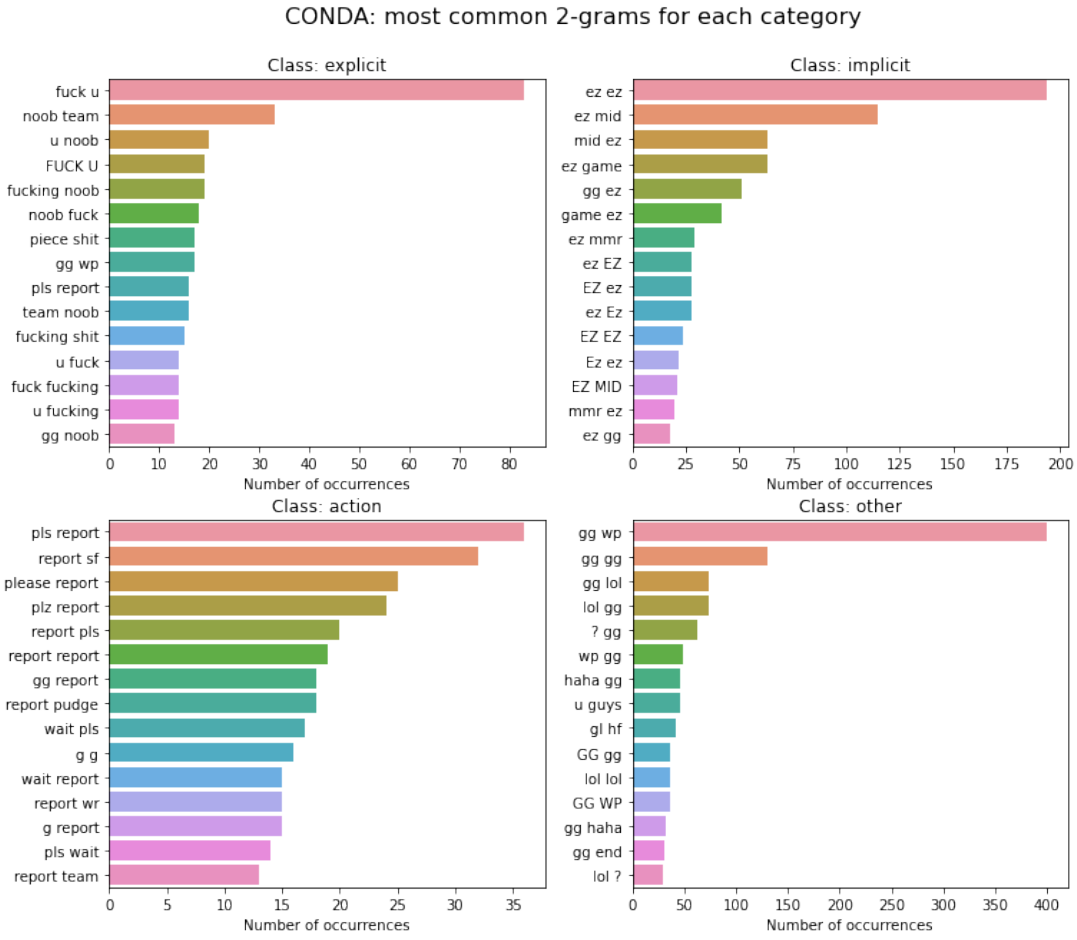


Figure 3.4: List of the most common 2-grams for each class.

Chapter 4

Research Gap and Contributions

As can be seen from the related work section, there are several gaps in the state of the art in the field of toxicity detection in games.
The work done by researchers is heterogeneous, focusing on different datasets, with no way to compare the different models. A **benchmark** would be necessary to compare the performances of different models.
The first objective of this thesis would then be to **compare the different techniques used in the literature on a single dataset**.
For this first task, it would be convenient to utilize the CONDA (DOTA2) dataset published by Weld et al. [48] As previously mentioned, the CONDA dataset classifies each utterance into 4 categories: Explicit, Implicit, Action, and Other.
In order to simplify the class system, Explicit and Implicit can become a "toxic" label, whereas Action and Other can be transformed into a "not toxic" label. However, this would mean losing a lot of important information. For example, utterances from the Implicit category will be expected to be misclassified more than the other classes by the models, as they are examples of covert toxicity. I will experiment using both the original set of labels and a binary one, as binary classification seems to be the most widely used in the literature.

The main reason for using this dataset is that it would be more convenient than annotating a new one. CONDA contains 45k utterances from 1.9k matches and is an English-only selection of messages from the largest videogame dataset currently available online. Furthermore, CONDA was already tested for intra-annotator agreement, which means that the labels are, to a certain degree, objective. Using this dataset would also increase its validity as a benchmark for the rest of the community. This dataset only utilizes text data, which means that multimodal approaches cannot be employed for the comparison.
The objective is to compare commercial models commonly used for toxicity detection (Perspective, Rewire), and open-source models widely used by the NLP community (BERT, Detoxify). The first research question can then be formulated as:

> *RQ.1: How will different toxicity detection models compare when tested on the same videogame dataset?*

This research question will be divided into two subquestions.

SQ1: How will pre-trained models work when they are not finetuned on the specific dataset?

This approach aims to be both a theoretical (shedding light on the viability of pre-trained models for automated content moderation in the context of games) and empirical

(a dataset containing all the new scores) contribution. As mentioned in the previous sections, there are multiple pre-trained models that have the objective of countering toxicity. These models will be tested on the CONDA dataset to see how they perform.
The first hypothesis concerning this experiment is:
H1: Since these models were trained on a large amount of toxic data, these models are expected to achieve good performance when analyzing toxic messages containing insults. However, since they were not trained on videogame jargon, I believe these models will not be able to generalize well on videogame slang.

Afterward, these models will be compared against a model architecture that can be finetuned, such as BERT (using Tensorflow). This specific architecture was chosen because it is among the most popular in NLP literature.
SQ2: How will finetuned BERT perform on the CONDA dataset when performing regression? For this first part of the thesis, a BERT model will be finetuned on the training set and a sigmoid layer will be added at the end of the dense layers (for regression). This added layer is necessary to standardize the inputs and generate a sequence of toxicity scores for each sentence. The answer to these questions will be measured using common classification metrics, such as accuracy or F1. After analyzing the dataset and computing the scores for each sentence, I aim to look more in-depth at the **disagreements in scores between models**. For this reason, I will generate confusion matrices and analyze the results. After computing the scores over the different datasets, the next goal would be analyzing and categorizing false positives and false negatives. False positives are utterances that are wrongly classified as toxic. False negatives, on the contrary, are toxic sentences that are mistaken as innocuous.

The second research question can then be formulated as:

> *RQ.2: What kind of in-game utterances are most commonly misclassified by models?*

This research question can be answered by observing the distribution of sentences being misclassified (original class they belong to, before the binarization) and the distribution of words belonging to misclassified sentences. It is interesting to observe, for example, whether context-specific words (belonging to game slang or common English words that have slightly different meanings in the game community) are commonly misclassified.

Finally, I am interested in observing whether BERT-like models will perform on the 4-class problem. The reason for this is that reducing the labels to their binary equivalents toxic/non-toxic comes at a cost: The Explicit and Implicit class both have their nuances and in a real-world scenario it would be important to be able to distinguish between the two. The same is not necessarily true for the Other and Action class, as they both contain innocuous sentences; nonetheless, this is still useful information being lost. For this reason, this research question will be formulated as:

> *RQ3: How does BERT perform in the multiclass detection task on the CONDA Dataset?*

To answer this, I will finetune BERT on the multiclass and also implement zero-shot learning. For what concerns the finetuning, I will compare base-BERT against tiny-BERT, a smaller implementation of the same model. At a base level, the models will probably have better performance in the multiclass scenario when compared to the binarized version of the problem. However, this is not necessarily true because the dataset is heavily skewed

with the "Other" category. Having only 2 classes partially compensated for this imbalance. Zero-shot learning is especially interesting because it is similar to the language APIs in terms of being a simple solution that does not require finetuning, but is inherently more flexible because the candidate labels can be changed based on the use case. Nonetheless, I expect this solution to not perform optimally, for the same reasons that were argued when discussing the pre-trained APIs.

Chapter 5

Methods

5.1 Models

In this section, I will provide an overview of the models that will be used during the course of the thesis. Perspective and Rewire are commercially available software that focus on toxicity in chats. These models were trained on a large amount of data, but since there is no way to finetune them on the dataset that will be utilized, I expect that these models will not be able to generalize on game language too well. However, they still are state-of-the-art in this kind of task, so they will be used as a baseline.

5.1.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model that uses a Transformer-based neural network architecture[15].

During fine-tuning, BERT can be adapted to specific NLP tasks by adding a task-specific output layer on top of the pre-trained model.
For regression tasks, such as predicting the numerical value of a continuous variable, BERT can be finetuned by adding a regression layer on top of the pre-trained model. The output layer consists of a fully connected layer followed by a linear activation function that produces a scalar output value. The final output of the model is the predicted value of the target variable. BERT has achieved state-of-the-art performance on several benchmark NLP datasets and has become a widely used tool in the NLP community. Over the course of this thesis, I will implement different flavors of BERT. The base architecture of a BERT Model, implemented on Tensorflow, can be seen in Figure 5.1. The preprocessing layer and the BERT_encoder layer vary based on the pretrained model being loaded from the Tensorflow Hub. The Dense layer will have a number of neurons equal to the number of classes being considered for the experiment. For the first part (the comparison between models), I will employ tinyBERT [45]. This model uses a standard BERT architecture but is significantly smaller. This model was chosen because it takes noticeably less time to train. Since the models employ scores, we will add a sigmoid layer for regression after the Dense layer with 2 neurons (discerning between Toxic and Non-toxic). It was also employed to highlight that good training data is more important than the complexity of the network. For the multiclass problem, I will compare TinyBERT against BERT with 12 transformer heads to see how complexity affects performance. In this case, the dense layer will have 4 neurons.

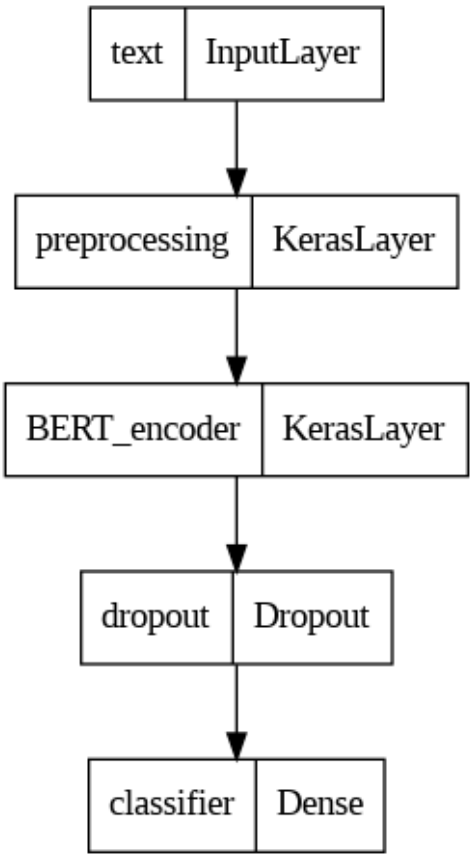


Figure 5.1: Example of BERT Architecture

Zero-Shot Classification

In the final part of the thesis, I will also use BERT to implement zero-shot classification to test the performance of the untrained models on the dataset. To perform zero-shot learning, it was decided to use the Hugging Face transformers library [49]. Hugging Face provides a set of pretrained models and an intuitive interface to leverage them for zero-shot learning. In principle, a model can be used to learn any set of candidate labels by simply using the following lines of code:

```
from transformers import pipeline
pipe = pipeline(model="facebook/bart-large-mnli")
pipe('The cake is a lie!',
    candidate_labels=["lie", "bakery", "videogames", "
       computer"]
    )
# output
>>> {'sequence': 'The cake is a lie!',
'labels': ["lie", "bakery", "videogames", "computer"],
'scores': [0.504, 0.479, 0.013, 0.005]}
```

For this part of the experiment, I will compare BERT, tinyBERT, and BART-large-mnli [29].

These models work by taking the candidate labels and constructing a hypothesis based on each one of them. For example, the candidate label "politics" is used to build the hypothesis "this text is about politics". For this reason, I tried various versions of the labels of the dataset to make them more meaningful. After various trials, the selected labels were: 'action' (A),'explicitly offensive' (E), 'implicitly offensive or sarcasm' (I), and 'game communication' (O).

5.1.2 Detoxify

Detoxify is a Python library that provides pre-trained models for detecting and mitigating toxic comments, hate speech, and other forms of harmful language in the text. The library is based on the Transformer architecture and is designed to be easy to use. It should be noted that the model is not meant to be finetuned and there are no built-in commands to "feed" the model a new dataset. Detoxify currently provides several pre-trained models. The main models are "original", "unbiased" and "multilingual". In this thesis, the "original" model was chosen because it seems the one targeted for the broadest use cases. After downloading the model, users can evaluate a sentence or a list of sentences with a simple call:

```
from detoxify import Detoxify
results = Detoxify('original').predict('The cake is a lie')
sentences = ['daje roma','ggez', 'tortellini']
results = Detoxify('original').predict(sentences)
```

Alternatively, it is also possible to use Command Line Interface (CLI) commands. The "results" object is a python list containing the scores, all in the range $[0, 1]$. The categories being considered are the following:

- Toxic
- Severe toxic
- Obscene
- Threat
- Insult
- Identity hate

For this study, we will focus on the Toxic label, as it is the most "meaningful" one.

5.1.3 Perspective

Google Jigsaw has released Perspective API [46], a large language model for content moderation. This API was invented for filtering comment sections and promoting discussion. It is currently used by large companies such as the New York Times to filter out toxic comments.
The architecture of the model is not public. However, it is known that the training data was labeled by human annotators, as can be seen from the explanation of their scoring system. Given a sentence as input, the call returns a number of scores. The model evaluates the sentence based on their Toxicity, which is defined as its "flagship score". In addition to that, the API also returns scores for the labels Severe Toxicity, Insult, Profanity, Identity attack, Threat, and Sexually explicit. These values range between 0 and 1 and are defined based on the judgment of human evaluators on the given metric. For example, a message such as "Shut up. You're an idiot!" would score a perfect 1.0 in the *insult* category, but only 0.15 in the *threat* one. This means that 100% of human reviewers would consider this sentence as an insult, but only 15% of them defines it as a threat. For our experiments, we will focus on the Toxicity score.

5.1.4 Rewire

Rewire is a language API whose objective is detecting abuse, hate, profanity, and sexually explicit language. The creators of the API claim that this model outperforms all competitors in terms of F1 when benchmarked on a series of hateful datasets. [1]
When given an API key, a user can send POST requests sending the key as a header and the text as a parameter. The API then returns a JSON object containing the original text, the request time, and a set of scores. The structure of the object showing the output scores can be seen in Figure 5.2. The scores are summarized as follows:

- Abuse indicates aggressive or insulting content.
- Hate refers to abuse towards a specific protected group or its members.
- Profanity score evaluates the presence of explicit words.
- Violent is defined as content that glamorizes violence.

[1] It should be noted that the company has been acquired and, as of 15/03/2023, the API is no longer operative.

- Sexually explicit is self-explanatory.
- Positive is a label that, opposite to the others, is used to highlight positive emotions.

Similarly to Perspective, the structure of the underlying model(s) is unknown.

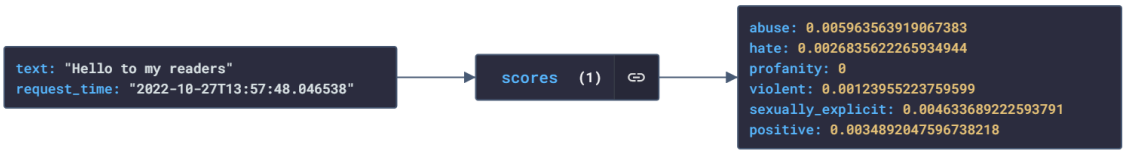


Figure 5.2: Example of a Rewire call. The request returns a "scores" object, containing the evaluation of the string passed as input.

Chapter 6

Results

6.1 RQ1

The first question that I aim to answer in this section is *How will different toxicity detection models compare when tested on the same videogame dataset?* The first step of the process is understanding how to bridge the gap between the labeling of the dataset and the way the different models perform.

The CONDA dataset has 4 utterance labels, 2 toxic (Explicit, Implicit) and 2 non-toxic (Action, Other). These labels were converted to their binary equivalent.
Detoxify, Perspective, and Rewire work similarly: given a string as input, these models return a set of scores comprised between 0 and 1, evaluating the sentence on different traits. As mentioned in the Methods section, for each of these models we will only consider the Toxicity score. Finally, BERT can be employed for regression by adding a sigmoid layer at the end of the to convert the classification confidence scores into a 0-1 probability range. That way, it is possible to generate scores that are similar to the ones of the other models, effectively standardizing the outputs of the models.

Using the "binarized" version of the CONDA dataset, I wanted to generate all the toxic scores and store them for easier comparison. Detoxify, Rewire, and Perspective were used to analyze all the utterances in the validation set. The scores were subsequently saved. Afterward, the BERT model was finetuned on the training set and evaluated on the validation set.

After these steps, I effectively generated a "new" dataset that contains a list of scores for each given sentence. This dataset contains 8974 sentences that have scores from all the models. An example of the core part of the dataset (excluding player names, timestamps, and other information) can be seen in Table 6.1.

This dataset is a contribution to the community, since the data can be used by researchers as a baseline to develop more accurate machine learning models. Furthermore, the data can facilitate studies on natural language processing by visualizing the flaws of

Utterance	**Class**	**Binary**	**P**	**R**	**D**	**B**
gg so bad beyond stupid	E	Toxic	0.788	0.954	0.894	0.961
GG NOOBS ez life ez noobs	I	Toxic	0.305	0.868	0.858	0.991
Report Luna thx .. AFK level 6 lol!	A	Non-toxic	0.112	0.001	0.004	0.013
Why are we all chatting?	O	Non-toxic	0.043	0.004	0.001	0.019

Table 6.1: 4 rows of the newly generated dataset containing the original labels, the binarized labels, and the scores from all the models.
P = Perspective, R = Rewire, D = Detoxify, B = BERT.

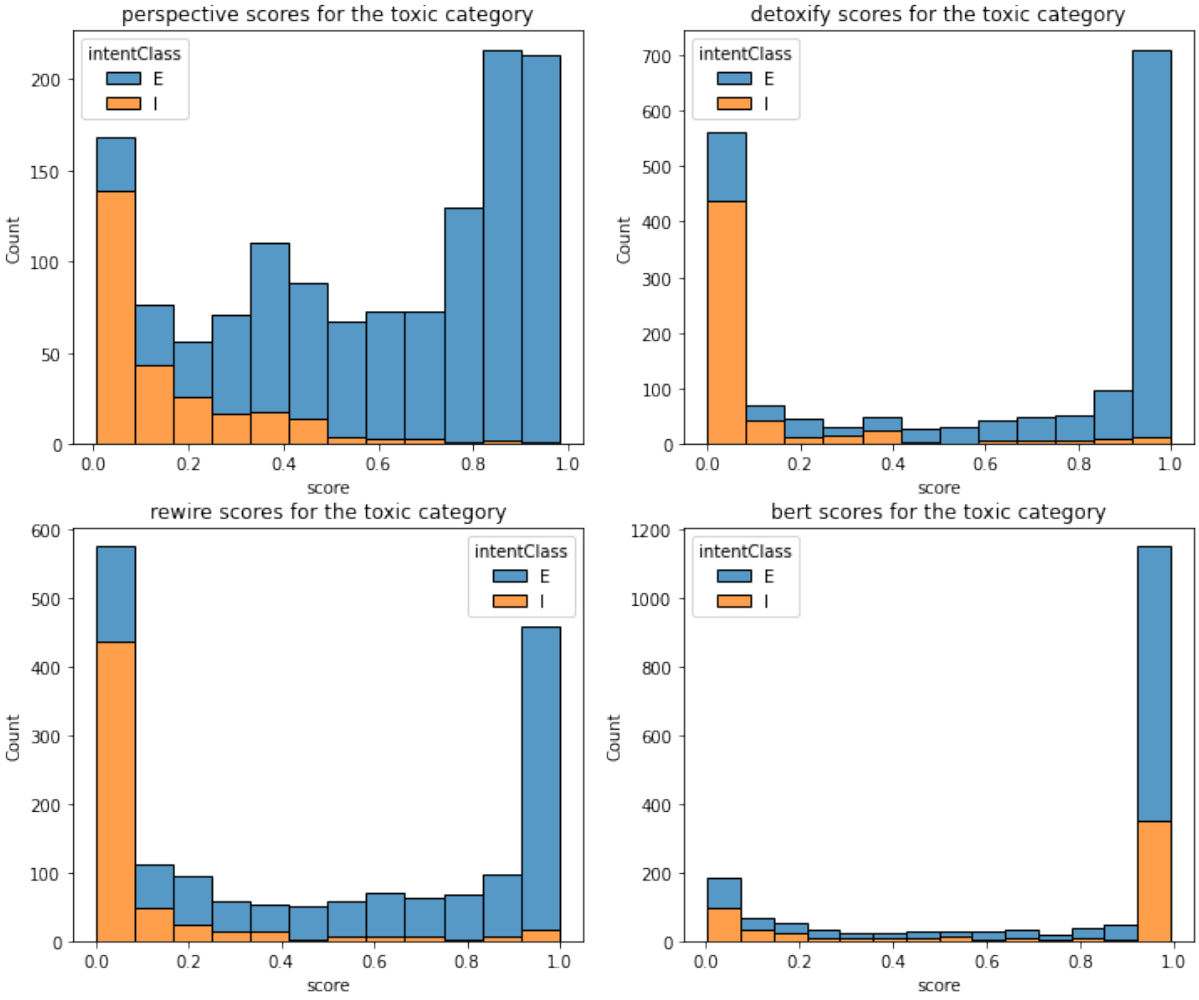


Figure 6.1: Scores for the toxic category for each model

each model. Finally, the data gathered in the following subsections will showcase the performance of these machine learning tools when it comes to moderating online content.

As mentioned earlier, the validation set was analyzed with 4 different models. A visualization of the scores can be seen in figure 6.2. It looks clear that a majority of the utterances have low scores. It is also clear that this is because of the class imbalance (Other being predominant).

By observing these scores, it is possible to gather some additional insight into the behavior of the different tools. For example, it is visible that the Implicit (I) class is classified as non-toxic by almost every model. This can be better gauged in Figure 6.1, which shows the score distribution of the toxic classes. it is visible that, for the APIs, most of the scores for the Implicit sentences are between 0 and 0.2.

Another interesting observation is that Perspective behaves differently from the other models, for a variety of reasons.

The first insight is gained from the actual amount of data shown in the plot. If one observes the y axes, there is an outlier in the number of elements for the collections of scores in the plot for Perspective. The model API, when asked to evaluate a message, first checks the language of the message. If it is not recognized, the API returns an 'Unknown Language' error. The CONDA dataset contains a large number of utterances that only

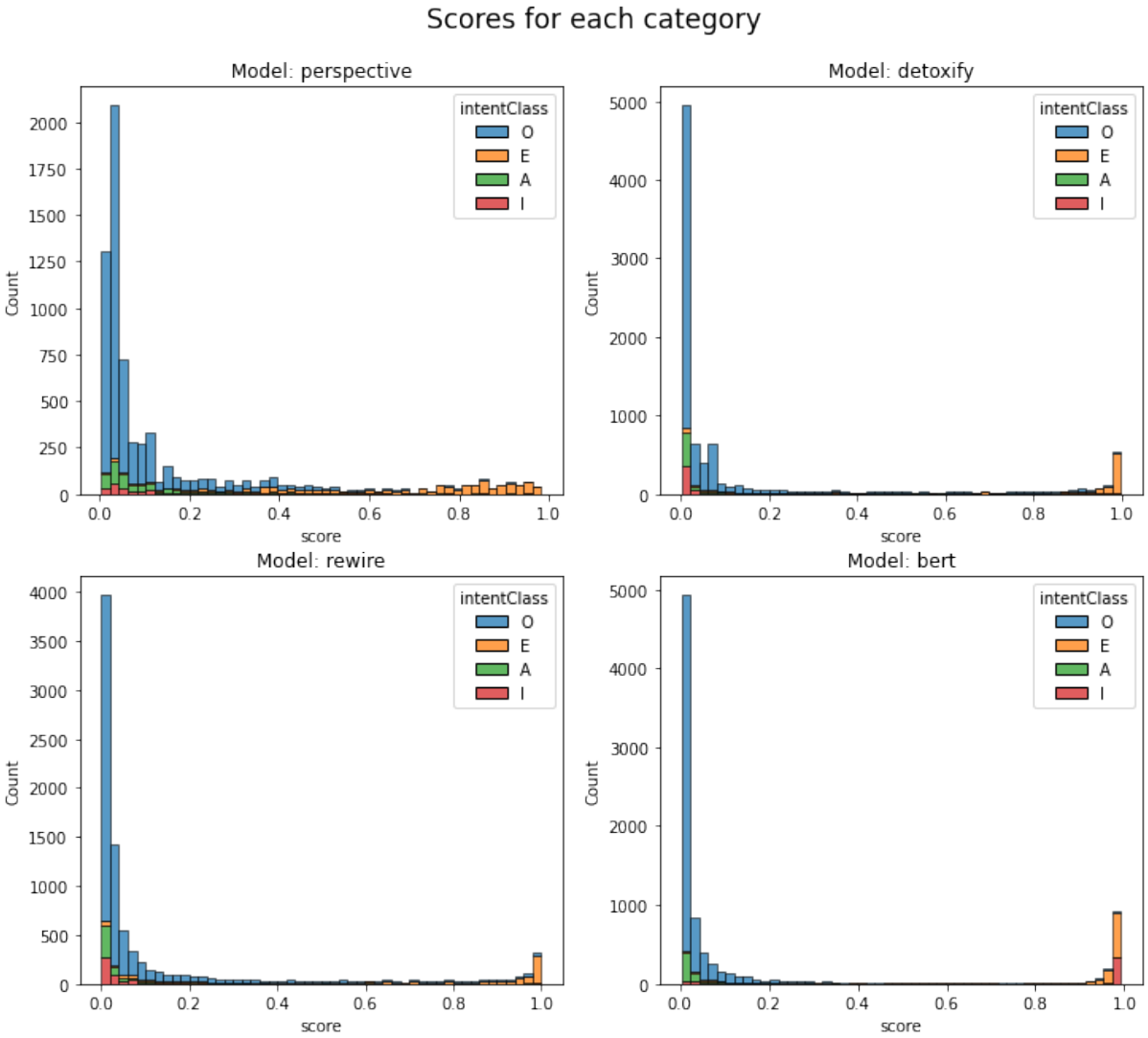


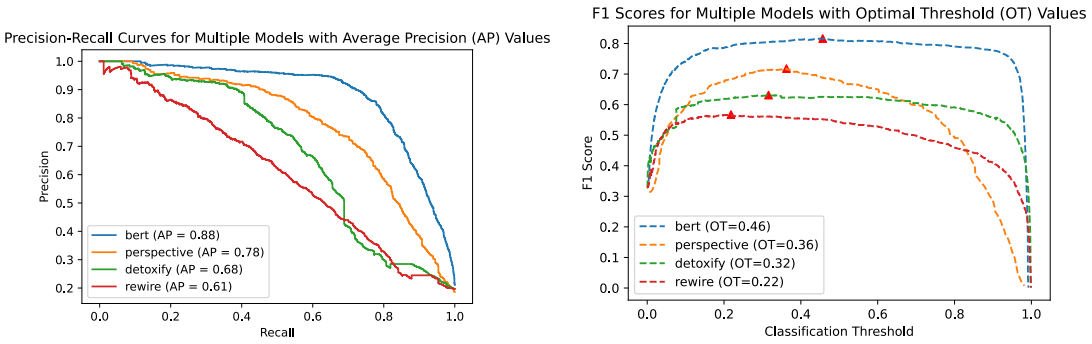
Figure 6.2: Visualization of the scores for each category.

contain game-specific slang, such as the words 'ez','gg' and so on.
This means that the model was not able to recognize a large number of utterances, as the model deemed they were not English.
More specifically, **1767** sentences were classified as unreadable (not in English). This means that Perspective was unable to classify **19.69%** of the utterances in the validation set. For clarity, these sentences have scores of -1 in the dataset. An example of this behavior can be seen in Table 6.2. More than half of the "implicit" entries were classified as unreadable.

In order to conduct a more quantitative analysis, it is useful to determine whether the utterances were classified correctly as toxic or non-toxic. The scores for each sentence are in the range (0,1). This means that accuracy, precision, recall and F-score fluctuate based on the choice of the threshold. Determining the optimal threshold is not trivial for a variety of reasons. For example, the scores for most models indicate the degree of confidence with which a given utterance can be classified as toxic or non-toxic. Perspective, however, uses the score as an additional way to convey information about the sentence. For example, a score of 0.5 means that "5 people out of 10" would find this sentence toxic.

Class	#Valid	#Invalid	Invalid rate (%)
Action	516	64	11.03
Explicit	1070	113	9.55
Implicit	271	311	53.44
Other	5350	1279	19.29

Table 6.2: Number of valid and invalid sentences on the dataset when analyzed using Perspective. The Invalid rate is calculated as $Invalid/(Valid + Invalid)$.



(a) Precision-Recall Curves.

(b) F1 Curves.

Figure 6.3: Precision-Recall Curves and f1 Curves for the different models. The plots also show the Average Precision and Optimal Threshold (Maximum f1-score).

This dataset has some other interesting characteristics. First of all, it is imbalanced, since the "Other" class is predominant. Moreover, I am interested in comparing multiple models at the same time. For all these reasons, these models will be compared using a Precision-Recall curve. The Precision-Recall curve is a useful metric when the classes are imbalanced. Intuitively, a high area under the curve represents high recall and high precision. Average Precision (AP) is a useful metric because it effectively summarizes the PR plot in a single number ($AP = \sum_n P_n * (R_n - R_{n-1})$). AP is then defined as the weighted average of the precision, where the weight is the difference between the recall of the nth threshold and from the previous one. Results can be seen in Figures 6.3a. From these results, it is visible that BERT outperforms the API models because the area under the curve is larger and it outperforms the AP of the second-best model by 10%. Moreover, using the Precision and Recall values, I plotted the values of the F1-scores ($F1 = 2 * \frac{P*R}{P+R}$). This curve allows for easy visualization of the threshold that maximizes F1. Results can be seen in 6.3b.

In the upcoming results, I will compare the models in two different ways: first, I will test them over a variety of thresholds to see which one resulted in the highest f-score, and the other at a fixed threshold of 0.5. Moreover, I will compute the confusion matrices per class based on the best values seen in the table. The confusion matrix for this can be seen in Figure 6.4. These matrices give us more insight into the type of misclassifications occurring in the dataset: the amount of false negatives and false positives is comparable, but considering the class imbalance it is clear that the main issue is that a lot of toxic sentences are being classified as non-toxic (false positives). Furthermore, Perspective has the lowest percentage of false positives. This suggests that the strategy of ignoring unclear

Thresholds	Perspective	Rewire	Detoxify	BERT
0.1	0.594	0.55	0.594	0.741
0.2	0.676	**0.565**	0.618	0.786
0.3	**0.712**	0.562	**0.629**	0.803
0.4	0.704	0.554	0.623	**0.812**
0.5	0.679	0.543	0.625	0.81
0.6	0.649	0.528	0.621	0.808
0.7	0.597	0.498	0.605	0.798
0.8	0.49	0.459	0.591	0.791
0.9	0.272	0.409	0.557	0.777

Table 6.3: f1-score per model, over different thresholds.

messages might be beneficial towards accuracy, although it should be remembered that this model ignored more than half of the messages contained in the implicit class.
These thresholds are selected *a posteriori* based on the best possible F1-score, but this would not be possible in a "real-world" scenario where there are no hand-labeled utterances. A different evaluation could then be conducted by choosing a unique threshold for all models. This approach can be seen in Figure 6.5. This Figure shows the confusion matrices for all models when the threshold is set to 0.5. This plot gives us more insight into the distribution of the misclassifications. Rewire has the worst f1-score (in accordance with Table 6.3), but the matrix gives the added insight that the number of false positives is higher than the number of true positives. BERT has a much lower ratio of misclassifications when compared to the other models, but still has a $\frac{FP}{FP+TP} \times 100 = \frac{419}{419+1346} \times 100 = 23.8\%$ false discovery rate.

To summarize, I generated a dataset containing a collection of binary scores. It was understood that the toxicity APIs have, on average, lower performance compared to finetuned small BERT. The next objective of this thesis is to find out more about the nature of the misclassifications. This subject will be covered in the next section.

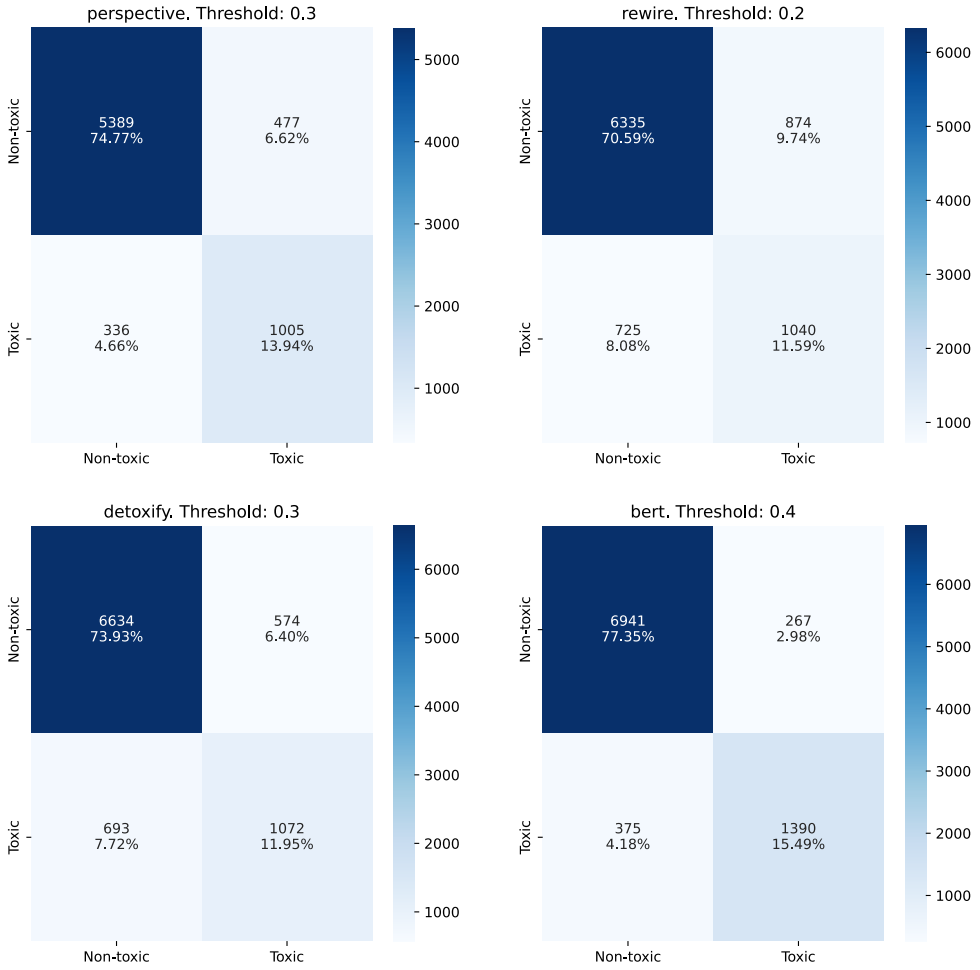


Figure 6.4: Best confusion matrices for each model.

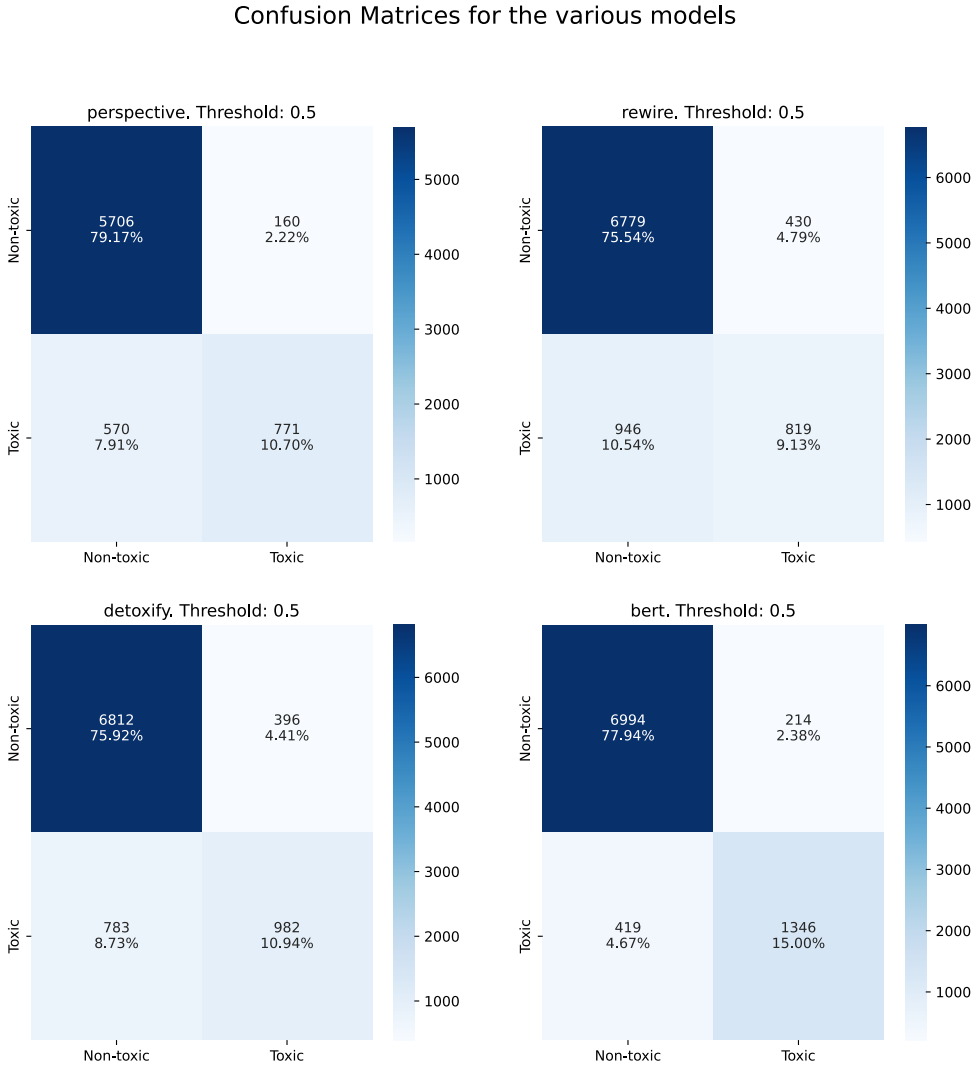


Figure 6.5: Confusion matrices for fixed thresholds

6.2 RQ2

In this section, I will answer the question: *What kind of utterances are most commonly misclassified?*

From the results shown in the previous section, it clearly emerges that all models have good performance on the Non-toxic class. However, especially for the APIs, the performance is lower for the Toxic class. To give more depth to the answer, refer to Table 6.4, where it is shown whether a label was correctly classified into the binary class. The table highlights the results for each of the 4 categories of the CONDA dataset. When keeping the label fixed (i.e. focusing on the explicit class), a sentence can only be a true positive (i.e. explicit (toxic) classified as toxic) or a false negative (i.e. explicit (toxic) classified as non-toxic). However, in a 4-class scenario, there is no way of knowing which class would be attributed to the misclassified sentence. For this reason, this value is defined as the recall per-class $\frac{TP}{TP+FN}$. The results confirm the intuition that the Implicit class is the hardest to classify, with the pretrained models scoring a maximum of **7.8%** on this class.
BERT scores the highest. However, the recall is only 67%. The hypothesis is that this result is a combination of the class imbalance and a loss of information that resulted from binarizing the classes of the dataset instead of keeping the 4 classes of the dataset.

Model	**Other**	**Action**	**Explicit**	**Implicit**
Perspective	**0.973**	**0.971**	0.707	0.052
Rewire	0.942	0.924	0.655	0.076
Detoxify	0.945	0.947	0.792	0.077
BERT	0.971	0.957	**0.807**	**0.672**

Table 6.4: Recall per class at Threshold 0.5

The Explicit class has comparable performances among the different classes, with BERT scoring the highest. Perspective also has a similar recall. Most of these performance problems are not really related to the "complexity" of sentences, but rather to the presence of words that the models can't recognize. The implicit class is dominated by the word "ez". The intuition is that the biggest impact on performance is due to the discrepancies from the models' original training data and the impossibility to finetune the APIs on game-specific data.

In this regard, it is possible to gather useful information from the distribution of game-specific slang. For this part of the experiments, it is useful to remember that the CONDA dataset contains, in addition to utterance labels, slot labels for each word: T (Toxicity), C (Character), D (Dota-specific), S (game Slang), P (Pronoun) and O (Other). For what concerns the analysis, the words belonging to the classes C, D, and S will be considered game slang. The subsequent step is checking the percentage of utterances containing any of these words. The results can be seen in Table 6.5.

Finally, it is interesting to observe which words are misclassified the most for each model. The process is the following: for each model, I will examine the binary classes. If a sentence is misclassified, the individual words of the sentence will be added to the list of misclassified words. After iterating over the entire class, the misclassified words are then sorted by order of occurrences. The 10 most frequent words for each class can be seen in

Class	Perspective		Rewire	
	Non-Toxic	**Toxic**	**Non-Toxic**	**Toxic**
Non-Toxic	5706 (3249) 56.94%	160 (131) 81.88%	6779(3447) 50.85%	430 (281) 65.35%
Toxic	570(411) 72.11%	771 (638) 82.75%	946(700) 74.00%	819 (664) 81.07%
	Detoxify		**BERT**	
	Non-Toxic	**Toxic**	**Non-Toxic**	**Toxic**
Non-Toxic	6812 (3471) 50.95%	396 (257) 64.90%	6994(3568) 51.02%	214 (160) 74.77%
Toxic	783 (580) 74.07%	982 (784) 79.84%	419(268) 63.96%	1346 (1096) 81.43%

Table 6.5: Confusion matrix for the models, showing the number of sentences in each class that were classified as toxic or non-toxic, as well as the number of sentences containing game slang in each class. The values in parentheses represent the number of sentences containing game slang out of the total number of sentences in that class.

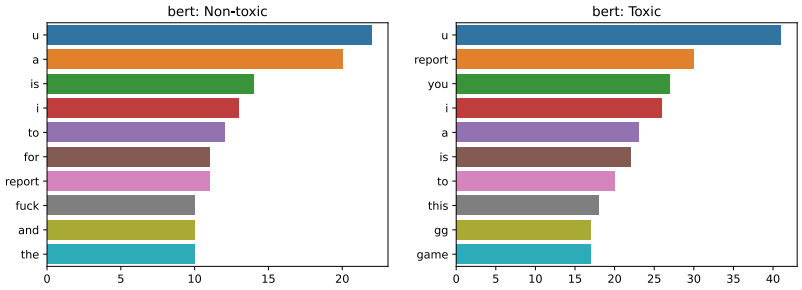
Figure 6.6.

It is clearly visible that most of the words are simply stopwords, such as the pronouns "i", "u", or "you". However, considering the false positives, it is interesting to observe that Perspective and Detoxify contain the words "Kill" and "Die", which were mentioned in the introduction as examples of words that have an extremely negative meaning in "normal" scenarios but are just normal game slang in the context of games. Furthermore, the most common words in false negative sentences contain a high amount of game slang. Confirming the hypotheses formulated in the previous Sections, the word "ez" is among the most misclassified by the APIs. This confirms the hypothesis that APIs struggle with understanding game-specific language.

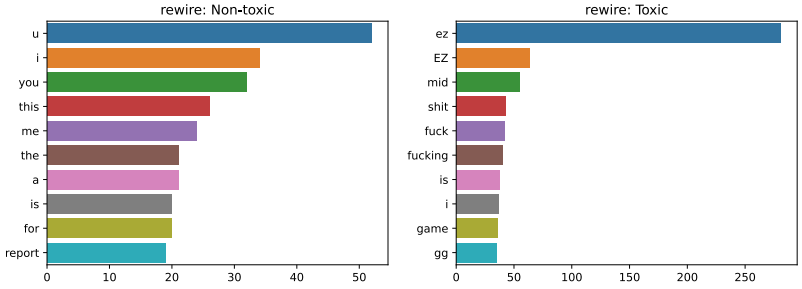
At the same time, it should also be noted that "ez", despite being sarcastic and an unsportsmanlike statement, is not a statement that a player should get banned for: the importance of detecting toxicity is extremely related to the use case, and while for some applications it would be important to detect all traces of toxicity (for example, to assess the state of wellbeing of the community), if the objective is only detecting really ban-worthy behavior, then the "implicit" class is mostly unimportant. This is, however, a matter of personal opinion. In principle, being limited to only detecting really harassing toxicity is a limitation of the language APIs that need to be taken into account when deciding to choose it for a real-world application.

6.3 RQ3

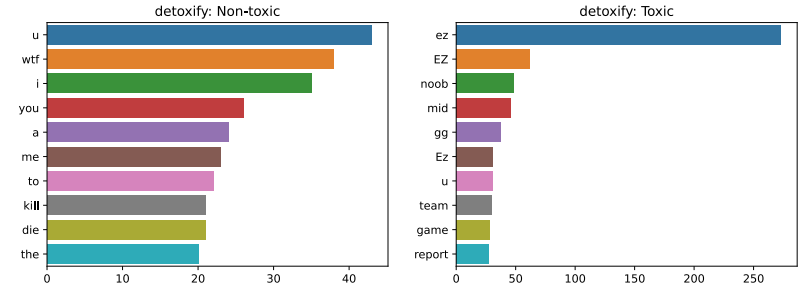
As already mentioned, reducing the labels to their binary equivalents toxic/non-toxic comes at a cost: The Explicit and Implicit class both have their nuances and in a real-world scenario it would be important to be able to distinguish between the two. For this reason, this section will focus on seeing how language models perform in the multiclass scenario. As of now, BERT was only used as a comparison term for the pretrained models. This model had better performances on the toxic class, although Detoxify (a model with



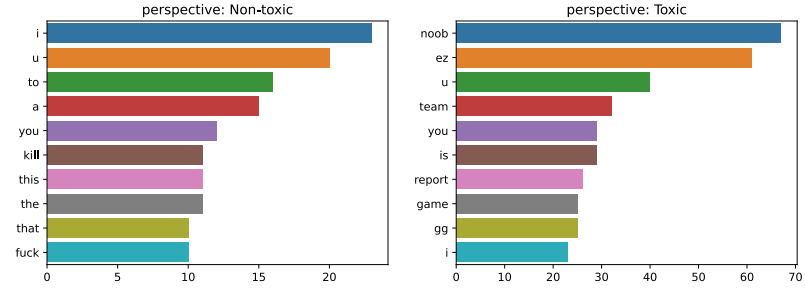
(a) Results for BERT



(b) Results for Rewire



(c) Results for Detoxify



(d) Results for Perspective

Figure 6.6: Frequency of misclassified words for each model. Non-toxic refers to the words belonging to non-toxic sentences that are actually wrongly classified as toxic (false positive). Toxic refers to words belonging to toxic sentences that are wrongly classified as Non-toxic.

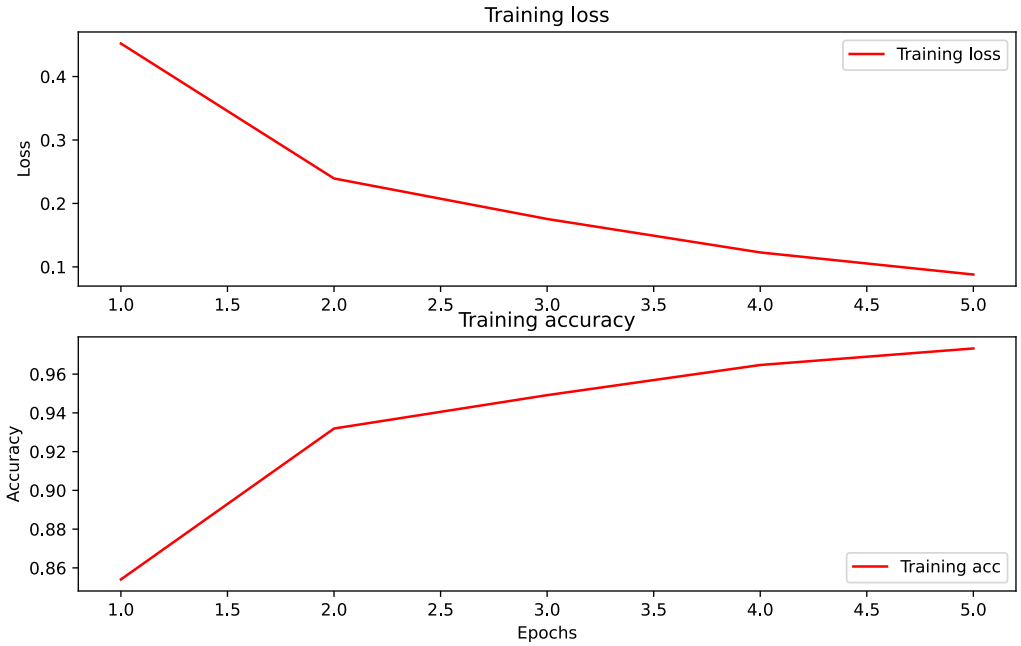


Figure 6.7: Loss and Accuracy for multiclass BERT while finetuning the model, over 5 epochs.

a similar structure) had a comparable performance for the Explicit class. However, this model can easily be extended to the multiclass scenario. There are multiple "flavors" of BERT. For example, previously I have used Tiny-BERT (L = 2, H = 128, A = 12). This model has 2 transformer layers, 128 hidden units, and 12 attention heads in each transformer layer. I will conduct a comparison between models with varying parameters. Finally, I will also perform an exploratory analysis to investigate whether this model could potentially perform reasonably well even without annotated data (zero-shot learning).

Comparing BERT performances

For the multiclass task, BERT was trained on the training set for 5 epochs. Accuracy and loss of the training can be seen in Figure 6.7.

This result was compared against Tiny-BERT. Results can be seen in Table 6.6. Both models have satisfying results in terms of both accuracy and F1. BERT has slightly better F1 and accuracy, but at the same time, training takes (roughly) 50 times longer. Both models were trained for 5 epochs. More detailed per-class statistics can be seen in Table 6.7. The Action class seems to be misclassified a lot more compared to the binary classification problem. However, this is not exactly true: earlier, the model was only determining whether the model was recognizing the sentence as "belonging to the non-toxic class". Most of the misclassification of the Action class in the 4-class problem belongs to the Other category (as is the case for every other class, due to the imbalance). For example, out of 580 Action sentences, Tiny-BERT classified 411 of them correctly, 145 as Other, and 24 as belonging to Explicit or Implicit. This means that roughly 96% of the sentences were deemed to be non-toxic, in line with the previous statistics.

Model	F1	Accuracy
tiny-BERT	0.804	0.902
BERT	0.844	0.921

Table 6.6: Accuracy and F1 Score for the Multiclass classification problem on the CONDA dataset. The results show how the finetuned models perform.

Model	Action	Other	Explicit	Implicit
TINY-BERT	0.75	0.94	0.81	0.75
BERT	0.79	0.95	0.87	0.76

Table 6.7: F1 Scores for the finetuned models, per class.

Zero-shot was performed using 3 different models: BERT, TINY-BERT, and BART-Large. Results can be seen in Table 6.8. It is visible that all the models performed extremely poorly (worse than random). It is hypothesized that the reason for this subpar performance is due to a variety of reasons. First of all, the name of the labels is vague. In the second place, the models were obviously bound to underperform on videogame slang.

Model	F1	Accuracy
tiny-BERT	0.0406	0.070
BERT	0.197	0.289
Bart-Large	0.145	0.175

Table 6.8: Results for the zero-shot classification problem on the CONDA Dataset.

Chapter 7

Discussion and Future Work

7.1 Discussion

Summary of results

In the Results section I focused on the detection of toxicity on the CONDA dataset. I focused on pretrained models that are commonly available online - namely Perspective, Detoxify, and Rewire - in order to assess their performance and compare them to a finetuned instance of BERT. This is, to date, the first instance of benchmarking toxicity models on a videogame dataset. I computed the different scores and generated a new dataset containing all the scores for 8974 sentences. Afterward, these models were compared by dividing the sentences into Toxic and Non-toxic. As seen in 7.1 BERT out-

Perspective	**Rewire**	**Detoxify**	**BERT**
0.679	0.543	0.625	**0.81**

Table 7.1: F1-scores for each model for a toxicity threshold of 0.5.

performed the pretrained models by a minimum of 13% F1 score for an arbitrary threshold of 0.5. Results were comparable for any other threshold. From the results I showed in the previous section, it was understood that the pretrained models did not perform optimally on the CONDA dataset. Further analysis showed that all the models had good performance on the non-toxic classes - performing with a minimum of 94% recall in the Other class and 92% in the Action Class. The models, however, had bad performances on the toxic classes. In particular, the pretrained models had issues with the Implicit class. In this instance, BERT recorded a 67.2% recall, while Detoxify, the second-best model in this class, only scored 7.7%. In the Explicit class, performance ranged from 65 to 80% recall.

Moreover, further analysis was conducted on the most commonly misclassified words for the models. In this instance, it was visible that the pretrained showed similar patterns: for example, common game slang - such as "ez" or "mid" were among the most frequent in the false negative category: sentences containing these words in the toxic category were often classified as non - toxic. Words that are common in game slang, but are associated with dark themes in other contexts - such as "kill" or "die" were commonly detected as toxic, even though they belonged in the toxic category.

Finally, I analyzed the performance of BERT on the 4-class dataset, with and without finetuning. It was discovered that - in the case of zero-shot learning - models performed poorly. This can be attributed to a series of factors, including the peculiar characteristics

of the dataset, the vague descriptive power of the labels, and the impossibility to give a description of the use case before starting the testing.

Limitations

Several limitations need to be considered for the study. First of all, the APIs proposed different types of scores. To simplify the comparison, only the "main" score for each model was considered.

In the second place, it should be noted that these results are valid for the CONDA dataset - but it is not sure whether they can be extended to different game datasets. CONDA uses DOTA2 chats, a hectic game where people need to type quickly, resulting in extremely short messages containing a lot of game slang. This does not necessarily apply to all games: for example, there are games that require fewer actions per minute, or where they can rely on voice chat. It would be interesting to generate a new game dataset to extend the study. Moreover, it is important to remember that DOTA2's API only allows obtaining data from the public chat. This means that it is not possible to read messages from the private "team" chat (which is usually used more often than the all-chat).

Moreover, it should be remembered that the labels of the dataset are generated by exclusion: the authors started by defining the Explicit Class as sentences that are outright offensive. Then, they said that the sentences in the Implicit class were offensive sentences that did not belong in the Explicit category. Action sentences are defined as sentences that are not I or E and contain action verbs. Other is defined as sentences that are not A, I, or E. It would be interesting to find a new labeling system that could be more informative.

7.2 Future Work

There are several ways this work can be extended. First of all, it was discovered that zero-shot learning is ineffective. However, fine-tuning the model on over 20.000 sentences is computationally expensive, and hard to expand to other contexts - as game developers would need to hand label a large amount of chat data to train a performing model. Moreover, in this specific scenario, it might not even be necessary to finetune on a large dataset, as most of the shortcomings of the language APIs could be easily addressed by giving some additional context. For example, it was observed that sentences containing specific keywords such as "ez", "gg", or "mid" have a higher probability of being toxic. For this reason, it would be extremely interesting to perform 1-shot and few-shot learning for the multiclass problem. It would be convenient to find a sweet spot between training set size and accuracy. Moreover, as mentioned in the limitations of the study, it would be interesting to generate data from a similar game, or from voice data, to see how the toxicity API would perform in a scenario where people are more verbose while playing.

Chapter 8

Conclusion

In this thesis, I have compared several state-of-the-art models for the detection of toxicity. I focused on pretrained models that are commonly available online - namely Perspective, Detoxify, and Rewire - to assess their performance and compared them to a baseline finetuned model of BERT.
This is, to date, the first instance of benchmarking these toxicity models on a videogame dataset. The different scores were computed and used to generate a new dataset containing all the scores for 8974 sentences. BERT outperformed the other models.
This work also showcases the importance of context: toxicity in the "traditional" sense is different from what is defined as toxicity in the gaming community, even though they both refer to bad behavior in online environments.
Moreover, this thesis highlights the need for flexible and easy-to-use models that can be adapted to the given environment.
The pretrained models, despite performing slightly worse than finetuned BERT, are still valuable tools, because it is not necessary to code to use these models, making them more accessible. BERT, however, allows users for more flexibility, along with the bonus of performance.

Overall, all these models are useful but have different use cases and users need to understand the pros and cons of each model before deciding which one to "pick". I believe this project helped shed light on this regard.

Bibliography

[1] Fair Play Alliance. *Disruption and Harms in Online Gaming. Framework*. 2020. URL: https://fairplayalliance.org/wp-content/uploads/2020/12/FPA-Framework.pdf.

[2] Aida Azadegan, Johann C. K. H. Riedel, and Jannicke Baalsrud Hauge. "Serious Games Adoption in Corporate Training". In: *Serious Games Development and Applications*. Springer Berlin Heidelberg, 2012, pp. 74–85. DOI: 10.1007/978-3-642-33687-4_6. URL: https://doi.org/10.1007/978-3-642-33687-4_6.

[3] Koray Balci and Albert Ali Salah. "Automatic Classification of Player Complaints in Social Games". In: *IEEE Trans. Comput. Intell. AI Games* 9.1 (2017), pp. 103–108. DOI: 10.1109/TCIAIG.2015.2490339. URL: https://doi.org/10.1109/TCIAIG.2015.2490339.

[4] Nicole A Beres et al. "Don't You Know That You're Toxic: Normalization of Toxicity in Online Gaming". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, May 2021. DOI: 10.1145/3411764.3445157. URL: https://doi.org/10.1145/3411764.3445157.

[5] Steven Bird and Edward Loper. "NLTK: The Natural Language Toolkit". In: *Proceedings of the ACL Interactive Poster and Demonstration Sessions*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 214–217. URL: https://aclanthology.org/P04-3031.

[6] Jeremy Blackburn and Haewoon Kwak. "STFU NOOB!" In: *Proceedings of the 23rd international conference on World wide web - WWW '14*. ACM Press, 2014. DOI: 10.1145/2566486.2567987. URL: https://doi.org/10.1145/2566486.2567987.

[7] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation". In: *J. Mach. Learn. Res.* 3.null (Mar. 2003), pp. 993–1022. ISSN: 1532-4435.

[8] Peter F. Brown et al. "Class-Based n-gram Models of Natural Language". In: *Computational Linguistics* 18.4 (1992), pp. 467–480. URL: https://aclanthology.org/J92-4003.

[9] Alessandro Canossa et al. "For Honor, for Toxicity". In: *Proceedings of the ACM on Human-Computer Interaction* 5.CHI PLAY (Oct. 2021), pp. 1–29. DOI: 10.1145/3474680. URL: https://doi.org/10.1145/3474680.

[10] Michelle Colder Carras et al. "Commercial Video Games As Therapy: A New Research Agenda to Unlock the Potential of a Global Pastime". In: *Frontiers in Psychiatry* 8 (Jan. 2018). DOI: 10.3389/fpsyt.2017.00300. URL: https://doi.org/10.3389/fpsyt.2017.00300.

[11] Luis Concepcion, Marilyn Nales-Torres, and Ana Rodriguez-Zubiaurre. "The Relationship between Videogame Use, Deviant Behavior, and Academic Achievement among a Nationally Representative Sample of High School Seniors in the United States". In: *American Journal of Educational Research* 4.16 (May 2017), pp. 1157–1163. DOI: 10.12691/education-4-16-6. URL: https://doi.org/10.12691/education-4-16-6.

[12] Mia Consalvo. "Confronting toxic gamer culture: A challenge for feminist game studies scholars". In: (2012). DOI: 10.7264/N33X84KH. URL: https://adanewmedia.org/2012/11/issue1-consalvo/.

[13] Christine Cook, Juliette Schaafsma, and Marjolijn Antheunis. "Under the bridge: An in-depth examination of online trolling in the gaming context". In: *New Media & Society* 20.9 (Dec. 2017), pp. 3323–3340. DOI: 10.1177/1461444817748578. URL: https://doi.org/10.1177/1461444817748578.

[14] Amanda C. Cote. ""I Can Defend Myself": Women's Strategies for Coping With Harassment While Gaming Online". In: *Games and Culture* 12.2 (2017), pp. 136–155. DOI: 10.1177/1555412015587603. eprint: https://doi.org/10.1177/1555412015587603. URL: https://doi.org/10.1177/1555412015587603.

[15] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[16] *Fair play alliance and ADL Rally Industry to combat hate and harassment in video games*. May 2022. URL: https://fairplayalliance.org/fair-play-alliance-and-adl-rally-industry-to-combat-hate-and-harassment-in-video-games/.

[17] Chek Yang Foo and Elina M. I. Koivisto. "Defining grief play in MMORPGs". In: *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology - ACE '04*. ACM Press, 2004. DOI: 10.1145/1067343.1067375. URL: https://doi.org/10.1145/1067343.1067375.

[18] Tommi Gröndahl et al. "All You Need is "Love"". In: *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*. ACM, Jan. 2018. DOI: 10.1145/3270101.3270103. URL: https://doi.org/10.1145/3270101.3270103.

[19] Xiaochuang Han and Yulia Tsvetkov. "Fortifying Toxic Speech Detectors Against Veiled Toxicity". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 7732–7739. DOI: 10.18653/v1/2020.emnlp-main.622. URL: https://aclanthology.org/2020.emnlp-main.622.

[20] Weszt Hart. *Player dynamics design: Looking behind the curtain*. May 2022. URL: https://www.riotgames.com/en/news/player-dynamics-design-looking-behind-the-curtain.

[21] Hossein Hosseini et al. *Deceiving Google's Perspective API Built for Detecting Toxic Comments*. 2017. DOI: 10.48550/ARXIV.1702.08138. URL: https://arxiv.org/abs/1702.08138.

[22] A.M. Kennedy et al. "Video Gaming Enhances Psychomotor Skills But Not Visuospatial and Perceptual Abilities in Surgical Trainees". In: *Journal of Surgical Education* 68.5 (Sept. 2011), pp. 414–420. DOI: 10.1016/j.jsurg.2011.03.009. URL: https://doi.org/10.1016/j.jsurg.2011.03.009.

[23] Yubo Kou and Xinning Gui. "Flag and Flaggability in Automated Moderation". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, May 2021. DOI: 10.1145/3411764.3445279. URL: https://doi.org/10.1145/3411764.3445279.

[24] Yubo Kou and Bonnie Nardi. "Regulating Anti-Social Behavior on the Internet: The Example of League of Legends". In: Jan. 2013. DOI: 10.9776/13289.

[25] Rachel Kowert. "Dark Participation in Games". In: *Frontiers in Psychology* 11 (Nov. 2020). DOI: 10.3389/fpsyg.2020.598947.

[26] Anti Defamation League. *Hate is no game: Harassment and positive social experiences in online games 2021*. Sept. 2021. URL: https://www.adl.org/hateisnogame.

[27] Alyssa Lees et al. "Capturing Covertly Toxic Speech via Crowdsourcing". In: *HCINLP*. 2021.

[28] Robert Lewington. *Being 'Targeted' about Content Moderation:Strategies for consistent, scalable and effective response to Disruption & Harm*. Apr. 2021. URL: https://fairplayalliance.org/wp-content/uploads/2022/06/FPA-Being-Targeted-about-Content-Moderation.pdf.

[29] Mike Lewis et al. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *CoRR* abs/1910.13461 (2019). arXiv: 1910.13461. URL: http://arxiv.org/abs/1910.13461.

[30] Marcus Martens et al. "Toxicity detection in multiplayer online games". In: *2015 International Workshop on Network and Systems Support for Games (NetGames)*. IEEE, Dec. 2015. DOI: 10.1109/netgames.2015.7382991. URL: https://doi.org/10.1109/netgames.2015.7382991.

[31] Lavinia McLean and Mark D. Griffiths. "Female Gamers' Experience of Online Harassment and Social Support in Online Gaming: A Qualitative Study". In: *International Journal of Mental Health and Addiction* 17.4 (July 2018), pp. 970–994. DOI: 10.1007/s11469-018-9962-0. URL: https://doi.org/10.1007/s11469-018-9962-0.

[32] Shane Murnion et al. "Machine learning and semantic analysis of in-game chat for cyberbullying". In: *CoRR* abs/1907.10855 (2019). arXiv: 1907.10855. URL: http://arxiv.org/abs/1907.10855.

[33] Lisa Nakamura. ""It's a Nigger in Here! Kill the Nigger!"". In: Dec. 2012. DOI: 10.1002/9781444361506.wbiems159.

[34] Joaquim A. M. Neto, Kazuki M. Yokoyama, and Karin Becker. "Studying toxic behavior influence and player chat in an online video game". In: *Proceedings of the International Conference on Web Intelligence*. ACM, Aug. 2017. DOI: 10.1145/3106426.3106452. URL: https://doi.org/10.1145/3106426.3106452.

[35] Rosario Ortega et al. "The Emotional Impact on Victims of Traditional Bullying and Cyberbullying". In: *Zeitschrift für Psychologie / Journal of Psychology* 217.4 (Jan. 2009), pp. 197–204. DOI: 10.1027/0044-3409.217.4.197. URL: https://doi.org/10.1027/0044-3409.217.4.197.

[36] Stephanie M. Ortiz. ""You Can Say I Got Desensitized to It": How Men of Color Cope with Everyday Racism in Online Gaming". In: *Sociological Perspectives* 62.4 (2019), pp. 572–588. DOI: 10.1177/0731121419837588. eprint: https://doi.org/10.1177/0731121419837588. URL: https://doi.org/10.1177/0731121419837588.

[37] Ryan Perry et al. "Online-only friends, real-life friends or strangers? Differential associations with passion and social capital in video game play". In: *Computers in Human Behavior* 79 (Feb. 2018), pp. 202–210. DOI: 10.1016/j.chb.2017.10.032. URL: https://doi.org/10.1016/j.chb.2017.10.032.

[38] Elizabeth Reid et al. ""Bad Vibrations": Sensing Toxicity From In-Game Audio Features". In: *IEEE Transactions on Games* (2022), pp. 1–1. DOI: 10.1109/tg.2022.3176849. URL: https://doi.org/10.1109/tg.2022.3176849.

[39] Paul Röttger et al. "HateCheck: Functional Tests for Hate Speech Detection Models". In: (2020). DOI: 10.48550/ARXIV.2012.15606. URL: https://arxiv.org/abs/2012.15606.

[40] Anna Schmidt and Michael Wiegand. "A Survey on Hate Speech Detection using Natural Language Processing". In: *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 1–10. DOI: 10.18653/v1/W17-1101. URL: https://aclanthology.org/W17-1101.

[41] Cuihua Shen et al. "Viral vitriol: Predictors and contagion of online toxicity in World of Tanks". In: *Computers in Human Behavior* 108 (July 2020), p. 106343. DOI: 10.1016/j.chb.2020.106343. URL: https://doi.org/10.1016/j.chb.2020.106343.

[42] Wessel Stoop et al. "Detecting harassment in real-time as conversations develop". In: *Proceedings of the Third Workshop on Abusive Language Online*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/w19-3503. URL: https://doi.org/10.18653/v1/w19-3503.

[43] John Suler. "The Online Disinhibition Effect". In: *CyberPsychology & Behavior* 7.3 (June 2004), pp. 321–326. DOI: 10.1089/1094931041291295. URL: https://doi.org/10.1089/1094931041291295.

[44] Joseph J Thompson et al. "Sentiment analysis of player chat messaging in the video game StarCraft 2: Extending a lexicon-based model". In: *Knowledge-Based Systems* 137 (Dec. 2017), pp. 149–162. DOI: 10.1016/j.knosys.2017.09.022. URL: https://doi.org/10.1016/j.knosys.2017.09.022.

[45] Iulia Turc et al. "Well-Read Students Learn Better: On the Importance of Pre-training Compact Models". In: *arXiv preprint arXiv:1908.08962v2* (2019).

[46] *Using machine learning to reduce toxicity online*. URL: https://www.perspectiveapi.com/.

[47] Bertie Vidgen et al. "Challenges and frontiers in abusive content detection". In: *Proceedings of the Third Workshop on Abusive Language Online*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 80–93. DOI: 10.18653/v1/W19-3509. URL: https://aclanthology.org/W19-3509.

[48] Henry Weld et al. *CONDA: a CONtextual Dual-Annotated dataset for in-game toxicity understanding and detection*. 2021. DOI: 10.48550/ARXIV.2106.06213. URL: https://arxiv.org/abs/2106.06213.

[49] Thomas Wolf et al. "HuggingFace's Transformers: State-of-the-art Natural Language Processing". In: *CoRR* abs/1910.03771 (2019). arXiv: 1910.03771. URL: http://arxiv.org/abs/1910.03771.