# Detecting Predatory Behaviour in Online Game Chats

Elín Rut Guðnadóttir, Alaina K. Jensen, Yun-Gyung Cheong, Julian Togelius, Byung Chull Bae, Christoffer Holmgård

Center for Computer Games
IT University of Copenhagen
Rued Langgaards Vej 7, Copenhagen, Denmark
Email: {elig, alaj, yugc, juto, byuc, holmgard@itu.dk}

**Abstract.** This paper describes a machine learning approach to detect sexually predatory behaviour in the massively multiplayer online game for children, *MovieStarPlanet*. The goal of this work is to take a chat log as an input and outputs its label as either the predatory category or the non-predatory category. From the raw in-game chat logs provided by *MovieStarPlanet*, we first prepared three sub datasets via extensive preprocessing. Then, two machine learning algorithms, naive Bayes and Decision Tree, were employed to model the predatory behaviour using different feature sets. Our evaluation has revealed that the proposed approach achieved high accuracies in detecting predatory chats.

**Keywords:** predator, text classification, preprocessing, chat, game data, NLP

## 1 Introduction

Social media is becoming increasingly prevalent today and children are frequent users of many different types of social media, such as online games [1]. Unfortunately, this can put them in a fragile position and enable others to take advantage of them. Due to this increased use of the Internet by minors, there is a growing need for sophisticated software which would allow children to be children and protect them at the same time.

To address this problem, our work is an investigation into effective methods for detecting predators and other rule-breakers in game-based chats, using recent data from the successful online game and community for children, *MovieStarPlanet*, collected in naturalistic settings.

While text classification algorithms have successfully been used for finding sexual predators in the past, our problem is different both because of the way we define sexual predators, and especially because we wish to do so for the context of an online game for children, using a dataset which has not been used before. This dataset is different from regular chatlogs in several ways: the game players sometimes chat "in character" as their respective movie star avatars, and frequently use a vocabulary unique to aspects of the game.

Morris and Hirst [2] define sexual predation as having two characteristics: "*age disparity*: a predator is an adult who chats with an underage individual" and "*inappropriate intimacy*: the adult must introduce or encourage intimate conversation." In our experiments, this definition is modified by the omission of the *age disparity* element, because of the context and circumstances of the game *MovieStarPlanet*.

Among the rules stated on *MovieStarPlanet*, is the rule "Don't write things that are sexually suggestive..." [1]. The rules also forbid the exchange of personal information such as addresses, phone numbers, or social network profiles. Because the rules specifically prohibit these behaviours, we define a *sexual predator* as

1. Anyone who initiates *sexually suggestive language*. This can be obvious as in "Let's have sex" or subtle as in "What does your underwear look like?"
2. Anyone who welcomes this type of language, and responds with similar language.
3. Anyone who tries to gain physical access to other users of the game (i.e. "Let's meet in real life").

In the context of this project, a user receives a predator (P) label based on this definition regardless of age, because the rules of the game strictly prohibit this type of language without respect to which person is using it. According to our definition, the term sexual predator is synonymous with the term rule breaker in the context of *MovieStarPlanet*.

### 1.1 Related Work

Because of privacy issues [1, 3] there is a general lack of publicly available data containing conversations between a sexual predator and real victim. However, chats containing conversations between a sexual predator and pseudo-victim, have been made available through nonprofit, volunteer based organisations, specifically perverted-justice.com. Perverted Justice (PJ)[2] trains adult volunteers to act as adolescents and go into chatrooms to talk with sexual predators until they incriminate themselves, at which point they are reported to the police. These chatlogs are available on PJ websites for everyone to use, and it is this data that most of the research in this area has been based on.

Pendar [4] used PJ data to train a model to discriminate between a victim and predator with good results: a $k$-NN classifier reached an $F$ measure of .943. Kontostathis et al [1] created rules to identify predatory text based on communication models. Their rule-based approach achieved 68% accuracy. They also experimented with machine learning algorithms, such as decision trees, which did not improve the accuracy.

Research done on pedophiles [5], shows that they often behave in distinct manner. Emotionally they are unstable and suffer from psychological problems.

[1] http://info.moviestarplanet.co.uk/terms-conditions.aspx
[2] http://www.perverted-justice.com/

It was on the basis of these findings that Bogdanova et al. [6] decided to try to distinguish predatory text from other text by using features such as emotional markers that could reveal predatory behavior. Other features to detect neuroticism level, fixated discourse, emoticons and imperative sentences were also used. A naive Bayes classifier was used, and it managed to discriminate between predatory text and other text with 94% accuracy. Peersman et al. [7] have however pointed out problems with the dataset used in the Bogdanova et al. experiment, which might undermine the results.

The most recent work done in the area of predator detection, is related to the PAN 2012 competition [3]. Sixteen teams contributed solutions to both detecting predators in the PAN 2012 dataset and finding the most predatory lines within the dataset. Villatoro-Tello et al. [8] was the team that was most succesful in detecting predators, but their system received an $F$ measure of .87. Their system consisted of two steps, in the first step dangerous conversations were found and in the second step, predator was distinguished from victim in the conversations previously found. No preprocessing or stemming was used, but conversations with less than 6 interventions per user were filtered out. Neural network classifier with binary weighting was used in both steps.

## 2 Our Approach

We consider the task of labeling predators in a chatlog to be a machine learning task, specifically a text classification task using supervised learning. The uniqueness of our work has much to do with our collaboration with *MovieStarPlanet* and the use of their data. This entails a unique set of challenges, which affect both the methodology and outcome of our experiments.

We found the style of chatting to be extremely different from other forms of written text, due to a very high level of misspellings, slang, grammatical errors, and seemingly meaningless symbols. Some of these characteristics are common in chat data as opposed to other online text data [6], but possibly even more so in the *MovieStarPlanet* dataset due to the young age (8 - 15 years) of the chat participants.

While the predator chatlogs we were given sometimes spanned a long period of time, the normal chatlogs we were provided with took place during a shorter time frame. We compensated for this by first manually filtering the predator chats, then limiting both predator and nonpredator chats to 15 minute sequences to make them uniform.

The nature of the game often calls for a language that may be similar to a predatory language, eg. the children can be in a relationship, they very often talk about dating, being single, looking for or having a boyfriend/girlfriend and loving their boyfriend/girlfriend. They also appear to frequently engage in family role play, e.g. “Pretend I am your dad/mom/sister/brother.” This sort of conversation

[3] http://www.uni-weimar.de/medien/webis/research/events/pan-12/pan12-web/authorship.html

within the normal chats formed a natural set of false positives to challenge the algorithm.

One of the greatest challenges is that the users of the game frequently try to circumvent the automatic safety nets in the game by using creative spelling and adding extra spaces, symbols, or line breaks when using words from *MovieStarPlanet*'s blacklist of words. To address this challenge, we have constructed a feature set which is designed to detect this type of behaviour.

## 3 Data

The dataset which was provided to us by *MovieStarPlanet* consists of all of the verbal communication from different users of the game, such as statuses, comments on videos and forum postings, as well as public and private chats from chatrooms and games. All userids and IP addresses were anonymized in this data, to protect the *MovieStarPlanet* users. We were provided with two types of datasets: predator data and non-predator data.

### 3.1 Non Predator Data

Normal chat data was given to us in 2 files by *MovieStarPlanet*, with each file containing approximately 65,000 rows, or 15 minutes of gameplay accross the entire UK site. One of those files was eventually used for Non Predator (NP) data in our training set (after extensive preprocessing), and the second file was not used as training data, but instead was saved for unlabeled testing. It is important to note that this normal chat data was all essentially unlabeled, meaning that *MovieStarPlanet* simply created a file containing 15 minutes of chat data across the entire UK site, and it was only an assumption that these files did not contain users which fulfilled our definition of a sexual predator.

### 3.2 Predator Data

Predator data was given as 1 file per predator (manually labeled by moderators), containing the messages typed by the predator, where the predator's anonymized id was found in the senderid column. These files also contained messages where the predator's id was found in the receiverid column, i.e. private messages which were sent to that user. This did not include messages typed within a group room where the predator was present, but only included situations where the predator's id was in the receiverid column. In total we received over 600 predator files from *MovieStarPlanet*, organised into the following categories:

- **0**: warning, the least severe punishment.
- **1**: 1 day IP address lockout.
- **3**: 3 day IP address lockout.
- **7**: 7 day IP address lockout.
- **99999**: permanent IP address lockout, the most severe punishment.

These categories were based on the punishments which these users received from *MovieStarPlanet* moderators. Each file contained full chatlogs (sometimes spanning several months) from users who had been punished for all types of offences, including sexual, as well as bullying, racism, or other types of rule disobedience. The different categories represented the seriousness of that offence. The files were not categorised according to the type of offence (sexual, bullying etc) because this is not something recorded by the *MovieStarPlanet* system at the time of punishment. The dataset therefore required extensive manual preprocessing in order to locate the sexual predators and separate them from the other types of rule-breakers, a process which was, of course, subject to human error. Once this was done, there were only 58 sexual predator files found out of the 600 original predator files.

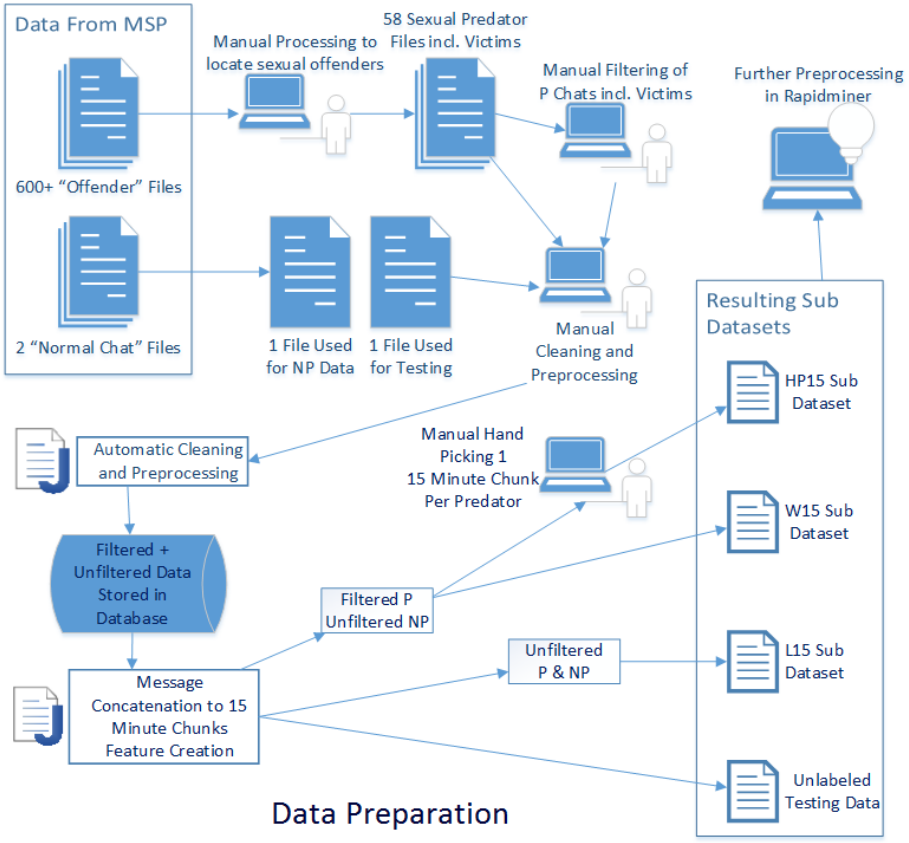


**Fig. 1.** An overview of the data preparation process, and how this preprocessing resulted in our final sub datasets. Note that after manual cleaning and preprocessing, there were two sets of predator data, one which contained entire unfiltered predator chats, and one which contained predator chats where nonpredatory lines had been removed.

## 4 Preprocessing

As shown in the figure 1, after manual processing to determine the type of offence, the 58 predator files, along with the normal chat files, were manually cleaned for a few minor errors, after which they were automatically cleaned and consolidated by a java program. After this step, both the P and the NP data was loaded into a database for convenient access. The initial testing which was performed on this data, including the full, unfiltered texts of the predators, had disappointing results. We hypothesised that this was due to the noise within the predator chats, caused by a high level of non predatory chatting (see table 2) in many of these files, with only short segments of chatting which was predatory in nature (see table 1). Therefore, we manually filtered each of the predator files and remove chatting which was deemed innocent and non predatory in nature.

**Table 1.** A list of some examples of predatory language.

| |
|---|
| who wants to have *** with me |
| can i fiddle with ur body? |
| what is your email |
| it with me i can feel it now uhh uhh uhh ohh harder HARDER |
| if u want to be my gf tell me where u live then |

**Table 2.** A list of some examples of language that would not lead to a sexual predator classification, though it might include some other type of offence.

| |
|---|
| Someone give me an auto and if u do i will give u one back! ;p |
| want to swap accounts |
| f* off other wise i will brske ur a* |
| ITS SPRING SUMMER HOLIDAYS BEACH ICE-CREAM AND I AM FREE NO SCHOOL FOR 2 WEEKS !!!!!!! |
| Nothing whats wrong with UR HAIR ITS WEARD |

The resulting filtered predator data was then subject to the same cleaning, preprocessing and consolidation as the unfiltered data had been, after which it was also loaded into a database (see figure 1). Thus we had a database containing both filtered and unfiltered predator data, and using another java program we were able to extract 3 different sub datasets from the database, drawing on both the filtered and the unfiltered P data, for use in classification experiments

### 4.1 Resulting Sub Datasets

Due to the fact that the predator and non predator data did not match in timespan due to the way it was collected, and contained a class imbalance problem,

it became necessary to create sub datasets before classification algorithms could be tested. In all sub datasets, a distribution of 20% predator data and 80% non predator data was used to reduce class imbalance, where the non predator data was randomly undersampled to achieve this distribution, and the predator data was prepared differently for each dataset as described below. To solve the mismatched timespan problem, predator data was always concatenated into 15 minute segments in each of the sub datasets, to match the 15 minutes of non predator chat.

When manually going through the files to determine which files represented sexual predators, it was noted that many of the violations which were recorded by the moderators were located near or towards the end of the file. This led to the hypothesis, that the last 15 minutes of a predator file might be the worst part because it represents the chatting which took place immediately before being "caught" by a moderator. Therefore, to test this hypothesis, a dataset was created which contained the last 15 minute segment of each predator chat, which constituted one tuple per user. This sub dataset was named **L15** (Last 15 Minutes of Unfiltered Predator Chats). For this dataset, the unfiltered data was used in order to avoid the situation where the last 15 minutes of the file might have been filtered out.

The filtered data was used to create 2 other sub datasets, which were named **W15** (Whole Filtered Predator Chats Divided into 15 Minute Segments) and **HP15** (Single Hand-Picked 15 Minute Segments of Filtered Predator Chats). For W15, we used the entire filtered predator chats, and when they were longer than 15 minutes, we simply divided them up into 15 minute segments, each of which then counted as a single tuple in the dataset. Though this dataset is the largest of all the sub datasets, it runs the risk of overfitting, because of the situation in which sometimes a single user was counted more than once, even as many as 35 times. For HP15, we wanted to eliminate the possibility of overfitting, so we manually picked only one 15 minute segment per user. When making this decision, we looked for the segments containing the most predatory language (see table 1), and also gave precedence to the longest chat segments.

## 5 Features

We used a combination of a classic bag of words (BoW) representation of the chat texts, with some standard sentiment analysis features, along with a set of features specifically designed to detect rule breaking behaviour in our dataset.

### 5.1 Bag of Words

As BoW has been established in countless previous works, e.g. in [2], as an effective feature for text classification, perfecting a BoW representation was our first priority as a feature in our classification process. In all our BoW representations, we used unigrams and bigrams, and pruned below the absolute of 9, in other

words, if a word or bigram was used in fewer than 9 tuples, this word/bigram was eliminated from the BoW. Stopwords were also eliminated.

As our initial attempts were not successful, we hypothesised that the cause of this could be the abundant misspellings in the dataset. This is because, with a BoW, each word or consecutive words in the dataset essentially becomes 1 feature, and the frequency of that word within the entire message becomes the value of that feature. This is only effective if the words are spelled in a uniform way, for instance if the word 'address' is spelled as 'adress,' 'addres,' 'adres' in addition to the correct spelling within the dataset, this creates 4 different features, when really it is only 1 word, and thus the BoW will be skewed.

Though some have hypothesised that misspellings can have an important meaning in the context of the chat and should not be corrected for that reason [4, 8], we conclude that, at least in our dataset, any potentially meaningful misspellings are dwarfed by the number of superfluous misspellings. As stated before, this is undoubtedly due to the young age of the chat participants in this case. Therefore, we used the Jazzy automatic spell checking API [4] to correct the message spelling, before creating a BoW from it.

### 5.2 Sentiment Features

5 simple features based on sentiment analysis were created, including positive, neutral and negative emoticons, and positive and negative sentiment scores. The emoticons were extracted from the data using regular expressions. In general, when classifying the emoticons, we used the rule that if any ambiguity was present, the emoticon was classified as neutral.

To obtain positive and negative sentiment scores, a simple approach to sentiment analysis was used, by searching through the message for words contained on the AFINN-111 wordlist [9] of labeled and scored sentiment words. The scores were collected and added together separately as a negative sentiment score and a positive sentiment score, to avoid the negative and positive numbers negating each others effect.

### 5.3 Rule Breaking Features

In addition to the BoW and sentiment features, a number of other features were created, which we will call *Rule Breaking Features*. The premises for creating these features, were the observations made on the dataset about the ways in which users try to avoid being caught when breaking the rules. Most of them involve avoiding the actual typing of blacklist words, in various ways. These rule breaking features can be further divided into blacklist features, and linguistic features, which will be described below.

[4] http://moderntone.blogspot.dk/2013/02/tutorial-on-jazzy-spell-checker.html

**Blacklist Features** *MovieStarPlanet* uses an extensive blacklist including alert words and blacklist words, with many spelling variations on each word. As *MovieStarPlanet* was kind enough to provide their full alert/blacklist for our project, it was possible in some cases to check for the number of alert or blacklist words within a text, and include this information in our features.

Alert words do not necessarily indicate bad behaviour, and may be said in the context of the game, but if too many are said, the moderator will be alerted. Blacklist words are however illegal in the context of the game, and are blocked. It is possible however in some cases to type a word which contains a blacklist word, surrounded by other letters or symbols, without that word being blocked. We therefore derived two features from the alert/blacklist: a count of alert words found in the text, and a count of how many times a blacklist word was enclosed in another word or phrase.

It is important to note that the data provided to us by *MovieStarPlanet* was unfiltered, meaning that in some cases it contained text which would be automatically blocked by the safety nets currently in place during realtime gameplay.

**Linguistic Features** The features in this category are all simple counts of some linguistic behaviour, with a logical reason behind each one based on rule breaking behaviour observed in the dataset.

- **One Letter Lines** - It has been observed that a user may avoid being caught typing an blacklist word, by typing one letter, hitting enter, typing the next letter, hitting enter, etc., until the full or partial blacklist word has been spelled out.
- **One Word Lines** - Similar to one letter lines, if a user wishes to type a forbidden phrase, this is sometimes done by typing one word at a time.
- **Lines** - The number of lines itself could possibly indicate suspicious behaviour.
- **Spaces** - The number of spaces can also be indicative. Sometimes users type blacklist words with spaces in between the letters in a word (e.g. "s e x") to avoid getting caught.
- **Non Letter Words** - Users will often try to avoid blacklist words by typing symbols or numbers in place of letters, for instance "s*x." This feature records a count of the words which contain symbols and/or numbers in addition to letters.
- **Consecutive Identical Letters** - Another rule breaking behaviour, is to use many consecutive identical letters inside a word, for example "seeeeex." It is assumed that more than 2 consecutive identical letters will never be necessary in the English language, therefore any word with more than two consecutive identical letters is counted in this feature.
- **Misspellings** - misspelling is also a technique for avoiding blacklist words, which has already been discussed, a misspelling count was included in our features.

## 6 Evaluation

A series of experiments were carried out under 10-fold cross validation with naive Bayes (NB) and J.48 decision tree (DT) classifiers from RapidMiner [10] with default parameter settings. Please refer to [11] and [12] for discussion on NB and to [13] for DT. We tested the performance of these two algorithms on three datasets (i.e., W15, L15, HP15) when using all features and also the combinations of several different feature sets, e.g., Sentiment and Rule Breaking, BoW and Blacklist.

In general, out of all three datasets, W15 performed best, with the best results being an $F$ measure of .90, 95.92% accuracy, and recall of .89 using DT on the Rule Breaking feature set. The L15 dataset performed worst in general, but like W15 performed best using Rule Breaking features on DT with an $F$ measure of .58, 85.5% accuracy and recall of .53. HP15 performed best using NB with BoW and Blacklist feature set, achieving an $F$ measure of .76, 90% accuracy, and recall of .82. The Sentiment features were the least useful feature set.

## 7 Conclusion

This paper presents a text classification method to detect sexually predatory behaviour. The article details a preprocessing strategy tailored to the *MovieStarPlanet* dataset, utilising principles such as focused resampling, and including the creation of 3 resulting sub datasets for testing different preprocessing strategies. We also created features unique to *MovieStarPlanet*: Bag of Words, sentiment features, and Rule Breaking Features. Rule Breaking features are designed to capture linguistic habits intended to avoid typing forbidden words. These features are more robust than a simple blacklist function, and can cross over into other NLP areas and games where this type of behaviour is common.

To fully exploit the bag of words features we used automatic spell checking, which improved the classification accuracy significantly. We hope that this result will provide useful insight for future research in a similar context. Given all of the above, our approach has achieved a classification result with nearly 96% accuracy when a decision tree algorithm on Rule Breaking features was employed.

For future work, a cleaner and more focused data is necessary, specifically more sexual predator files are needed. Current features could also be improved, starting with a better spell checker, that would improve the BoW features further. It would also be interesting to add as a feature, information about the recipient of the message, whether it was sent in a private, public or other setting. Finally, it would also be interesting to incorporate our system, with the system already in place at *MovieStarPlanet*, where a history of some users is recorded.

## 8 Acknowledgement

This work has been supported in part by the EU FP7 ICT project SIREN (project no: 258453).

## References


1. Kontostathis, A., Edwards, L., Leatherman, A. In: Text Mining and Cybercrime. John Wiley & Sons, Ltd (2010) 149–164
2. Morris, C.: Identifying online sexual predators by svm classification with lexical and behavioral features. (2013)
3. Inches, G., Crestani, F.: Overview of the international sexual predator identification competition at pan-2012. In: CLEF (Online Working Notes/Labs/Workshop). (2012)
4. Pendar, N.: Toward spotting the pedophile telling victim from predator in text chats. In: Proceedings of the International Conference on Semantic Computing. ICSC '07, Washington, DC, USA, IEEE Computer Society (2007) 235–241
5. McGhee, I., Bayzick, J., Kontostathis, A., Edwards, L., McBride, A., Jakubowski, E.: Learning to identify internet sexual predation. International Journal of Electronic Commerce **15**(3) (2011) 103–122
6. Bogdanova, D., Rosso, P., Solorio, T.: On the impact of sentiment and emotion based features in detecting online sexual predators. In: Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis. WASSA '12, Stroudsburg, PA, USA, Association for Computational Linguistics (2012) 110–118
7. Peersman, C., Vaassen, F., Van Asch, V., Daelemans, W.: Conversation level constraints on pedophile detection in chat rooms. In: CLEF (Online Working Notes/Labs/Workshop). (2012)
8. Villatoro-Tello, E., Juárez-González, A., Escalante, H.J., Montes-y Gómez, M., Pineda, L.V.: A two-step approach for effective detection of misbehaving users in chats. In: CLEF (Online Working Notes/Labs/Workshop). (2012)
9. Nielsen, F.Å.: A new anew: Evaluation of a word list for sentiment analysis in microblogs. arXiv preprint arXiv:1103.2903 (2011)
10. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: Yale: Rapid prototyping for complex data mining tasks. In Ungar, L., Craven, M., Gunopulos, D., Eliassi-Rad, T., eds.: KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (August 2006) 935–940
11. McCallum, A., Nigam, K., et al.: A comparison of event models for naive bayes text classification. In: AAAI-98 workshop on learning for text categorization. Volume 752., Citeseer (1998) 41–48
12. Han, J., Kamber, M., Pei, J.: Data mining: concepts and techniques. Morgan kaufmann (2006)
13. Suh, S.C.: Practical applications of data mining. Jones & Bartlett Publishers (2012)