UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

How to take advantage of behavioral features for the early detection of grooming in online conversations

Daniela Fernanda Milón Flores

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP
SÃO CARLOS

Daniela Fernanda Milón Flores

How to take advantage of behavioral features for the early detection of grooming in online conversations

Dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Master in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Robson Leonardo Ferreira Cordeiro

USP – São Carlos
March 2022

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

```
        Milón Flores, Daniela Fernanda
637h       How to take advantage of behavioral features for
        the early detection of grooming in online
        conversations / Daniela Fernanda  Milón Flores;
        orientador Robson Leonardo   Ferreira Cordeiro. --
        São Carlos, 2022.
           125 p.


           Dissertação (Mestrado - Programa de Pós-Graduação
        em Ciências de Computação e Matemática
        Computacional) -- Instituto de Ciências Matemáticas
        e de Computação, Universidade de São Paulo, 2022.


           1. Early text classification. 2. Classification
        with partial information. 3. Behavioral features.
        4. Online detection of grooming. I. Ferreira
        Cordeiro, Robson Leonardo  , orient. II. Título.
```

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Daniela Fernanda Milón Flores

Como aproveitar características comportamentais para a detecção precoce de assédio sexual em conversas on-line

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestra em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Robson Leonardo Ferreira Cordeiro

USP – São Carlos
Março de 2022

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank Professor Dr. Robson L. F. Cordeiro, who has guided me during these two years and from whom I have learned a lot. I trust I can use his teachings in the future. I thank my parents, Rita Flores and Carlos Milon. They made me feel their company despite being away for several months. All this effort is for you. Also thank to my grandparents, who are my angels. To my friends, who made me feel at home. And to Rita Sant'Ana, my first friend from Brazil, whom I now consider part of my family.

Finally, I would like to thank the financial support from the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Finance Code 001, the Fundação de Amparo à Pesquisa do Estado de São Paulo - Brasil (FAPESP) – 2021/05623-2, 2020/07200-9, 2018/05714-5 and 2016/17078-0, and; the Conselho Nacional de Desenvolvimento Científico e Tecnológico – Brasil (CNPq).

“Concedei-nos, Senhor,
a serenidade necessária para aceitar as coisas que não podemos modificar,
coragem para modificar aquelas que podemos,
e sabedoria para distinguir umas das outras.”
(Oração da Serenidade)

RESUMO

FLORES, D. M. **Como aproveitar características comportamentais para a detecção precoce de assédio sexual em conversas on-line**. 2022. 122 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2022.

A detecção de comportamentos de assédio sexual em conversas online tornou-se um problema crescente devido ao grande número de plataformas de mensagens que as crianças e os jovens utilizam actualmente. O maior inconveniente é a falta de ferramentas centradas na prevenção automática deste risco. Este documento propõe sete características comportamentais a serem utilizadas para a detecção precoce do assédio. É realizado um estudo detalhado para compreender os antecedentes que permitem que estas características contribuam para tarefas de classificação precoce. Além disso, introduzimos o framework Behavioral Feature - Profile Specific Representation (BF-PSR) como uma extensão do bem conhecido framework Profile Specific Representation (PSR) para empregar correctamente as características comportamentais propostas. Os resultados experimentais revelam que a nossa proposta supera todos os métodos concorrentes e obtém um desempenho de vanguarda na área da detecção precoce de assédio. Especificamente, a nova estrutura BF-PSR atinge um ganho de mais de 40% em eficácia sobre cinco concorrentes quando apenas 10% do conteúdo das conversas SGD está disponível, mostrando assim uma vantagem substancial para permitir a detecção precoce do asseio; além disso, mantém um ganho de eficácia semelhante à medida que chegam mais dados. Ademais, tanto quanto sabemos, este é o primeiro trabalho a empregar características comportamentais para a detecção precoce do assédio. Por outro lado, montamos dois novos conjuntos de dados chamados PJZ e PJZC para mitigar a falta de dados na área de detecção do asseio. Ambos conjuntos estão publicamente disponíveis para download, com o objectivo de fomentar a continuação da investigação. Experimentos adicionais revelam que a nossa estrutura BF-PSR supera todos os métodos actuais no processamento destes novos conjuntos de dados.

Palavras-chave: Classificação precoce de texto, classificação com informação parcial, características comportamentais, detecção de assédio sexual on-line.

ABSTRACT

FLORES, D. M. **How to take advantage of behavioral features for the early detection of grooming in online conversations**. 2022. 122 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2022.

Detecting grooming behavior in online conversations has become a growing problem due to the large number of messaging platforms that children and young people use nowadays. The biggest drawback is the lack of tools focused on the automatic prevention of this risk. This paper proposes seven Behavioral Features (BFs) to be used for early grooming detection. A detailed study is conducted to understand the background that allows these features to contribute to tasks of early classification. Besides, we introduce the Behavioral Feature - Profile Specific Representation (BF-PSR) framework as an extension of the well-known Profile Specific Representation (PSR) framework to properly employ the proposed behavioral features. Experimental results reveal that our proposal outperforms all the concurrent methods and obtains state-of-the-art performance in the area of early grooming detection. Specifically, the new BF-PSR framework achieves a gain of more than 40% in effectiveness over five competitors when only 10% of the SGD conversations' content is available, thus it shows a substantial advantage to allow the early detection of grooming; besides, it maintains a similar gain in effectiveness as more data arrives. Furthermore, to the best of our knowledge, this is the first work to employ behavioral features for the early detection of grooming. On the other hand, we have assembled two new datasets called PJZ and PJZC to mitigate the lack of data in the grooming detection area. Both sets are publicly available for download aimed at fostering further researches. Additional experiments reveal that our BF-PSR framework outperforms all state-of-the-art methods when processing these new datasets.

Keywords: Early text classification, classification with partial information, behavioral features, online detection of grooming.

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

BF-PSR	Behavioral Feature - Profile Specific Representation
BFs	Behavioral Features
BoC	Bag of Centroids
BoW	Bag of Words
CNN	Convolutional Neural Network
ENB	Early Naive Bayes
ETC	Early Text Classification
FFNs	Feed Forward Networks
FN	False Negative
FP	False Positive
G	Groomer
IDF	Inverse Document Frequency
IWF	Internet Watch Foundation
LR	Logistic Regression
ML	Machine Learning
MLP	Multilayer Perceptron
MulR	Multi-resolution Representation
NCMEC	National Center for Missing and Exploited
NG	Non-groomer
NSPCC	The National Society for the Prevention of Cruelty to Children
PJ	Perverted Justice Foundation
PSR	Profile Specific Representation
RF	Random Forest
SGD	Sexual Groomer Detection
SLSS	Social Live Streaming Service
SVMs	Support Vector Machines
TF	Term Frequency
TF.IDF	Term Frequency . Inverse Document Frequency
TN	True Negative
TP	True Positive
TVT	Temporal Variations of Terms
W2V	Word to Vector

CONTENTS

CHAPTER 1

INTRODUCTION

Nowadays it is difficult to perform activities without access to the Internet; either for work, study, or entertainment purposes, this tool has become a fundamental part of our day-to-day life. Children and young people are not isolated from this lifestyle. They have access to all the goodness of the Internet, such as the rapid search of information, long-distance communication, multiple sources of recreation, etc. However, in the same way that they take advantage of these benefits, others can make use of this network to perform activities that put children's integrity at risk, as, online groomers. The National Society for the Prevention of Cruelty to Children (NSPCC)[1] defines groomers as adults who build an emotional and trusting relationship with children through an online platform to exploit and abuse them. Recently, the number of attacks under this approach has been growing, and, according to a statement made by the National Center for Missing and Exploited (NCMEC)[2], only in 2020 their online system CyberTipline has received approximately 21.7 million reports of child sexual exploitation. In the same year, the analysts at the Internet Watch Foundation (IWF)[3] processed a record of 299,619 reports, which represents an increase of 16% when compared with 2019.

It is also important to mention that according to the NSPCC, platforms such as Facebook, Instagram, WhatsApp, Snapchat, and other famous social media networks formed part of the online places with the highest number of groomer attacks in 2021. Besides, they found out that one in five victims of online grooming is under 11 years old and they explained, through a detailed article[4], how anxiety, depression, and guilt can be some of the effects of grooming on children and young people.

Nevertheless, although this problem results extremely alarming, it is surprising that there are only a few works focused on the preventive scenario of early detection of online grooming, i.e.,

[1] <https://www.nspcc.org.uk>
[2] <https://www.missingkids.org/gethelpnow/cybertipline>
[3] <https://www.iwf.org.uk>
[4] <https://www.nspcc.org.uk/what-is-child-abuse/types-of-abuse/grooming/>

tools dedicated to prevent an attack by detecting initial attempts as early as possible. In fact, most of the existing tools that detect grooming consider a forensic scenario, i.e., these tools can only be used when the attack has already occurred (ESCALANTE *et al.*, 2017). Accompanied by this drawback, there is a lack of datasets available for use in research due to the obvious sensitiveness of the information contained in conversations between a child and a groomer. In this way, the corpus Sexual Groomer Detection (SGD) is the most popular and used dataset in the area of grooming detection. Due to its characteristics, e.g., easy access, labeled, well-structured, composed of a large number of conversations, etc., it has become the only dataset employed by tools that work in a preventive scenario, which converts it into a valuable resource in the area of early grooming detection. Unfortunately, preventive tools dedicated to the early detection of grooming apply filtering techniques to soften the imbalance of the SGD data; as a consequence, they ignore information that may be relevant to the detection of groomers and present masked results because they do not report as errors the groomer conversations that were filtered out from the analysis. Also, since these methods were only tested with filtered data, there is a possibility that they do not perform well with a large amount of disproportionate information, which is the scenario expected in most real applications. Another notable limitation is the absence of Behavioral Features (BFs) in preventive tools. BFs are characteristics based on the behavior of the person, e.g., signature, emotions, voice, etc. When applying in text classification tasks, the goal is to understand how a user employs a word and not focus on which words the user employs (MORRIS; HIRST, 2012). To the best of our knowledge, none of these tools employ BFs to detect grooming behavior in online conversations, which is a gap in the literature since their importance is confirmed by the forensic methods that employ these features.

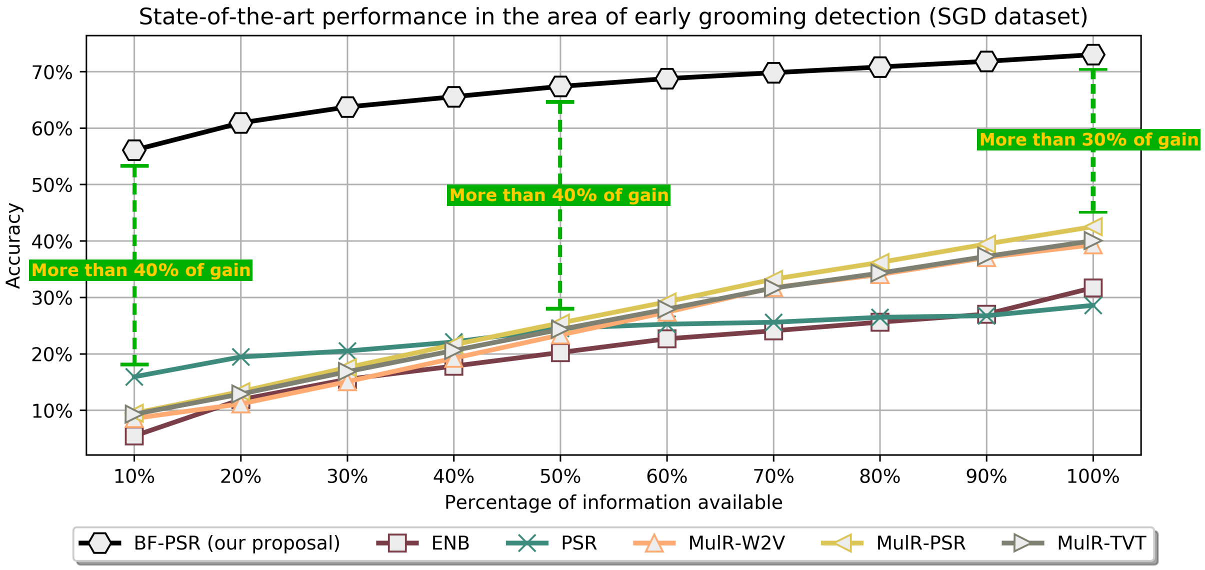
As we have observed, there is a great necessity to develop preventive tools capable of detecting groomer behavior effectively in a large amount of data. In this MSc work, we aim to confront this problem by exploring the following hypothesis:

> **Hypothesis**: The proper use of Behavioral Features contributes to the early detection of grooming in a large set of online conversations.

To validate our hypothesis, a study is conducted to understand the background that allows these BFs to contribute to tasks of early classification. Besides, we introduce our proposal, the Behavioral Feature – Profile Specific Representation (BF-PSR) framework to properly integrate the mentioned features into a vector representation that highlights the most relevant patterns of a grooming conversation. Finally, we compare our proposal with the most recent state-of-the-art methods. Following, we present our main contributions in detail:

C1 **Behavioral features for the early detection of online grooming:** A set of seven Behavioral Features (BFs) is proposed to detect grooming behavior in online conversations. Specifically, our proposed features refer to the number of participants in a conversation,

Figure 1 – Our framework BF-PSR is compared against four concurrent methods in the area of early detection of grooming, i.e., the original PSR and the three variants of method MulR. The plot reports the accuracy of results achieved by each method when different percentages of text are available for processing. As it can be seen, our proposal outperforms all the concurrent methods obtaining a gain in accuracy of more than 40% when only 10% of information is available, which is very attractive for early detection. We also obtained a gain of more than 40% in accuracy with 50% of the information available and a gain of more than 30% with the whole SGD dataset. Therefore, our proposal obtained state-of-the-art results in the area of early grooming detection.



emoticons, correctly-spelled words, sexual topic words, the time when a conversation starts, intervention words per user, and sentiment and emotional markers. In addition, a detailed study is conducted to understand the reasons why each feature contributes to the task of early detection of grooming.

C2 **PSR*:** In order to represent a large set of unbalanced conversations in a non-sparse and low-dimensional vector space, we propose a variation of the original PSR algorithm (ESCALANTE *et al.*, 2017) which we name as PSR*. This new algorithm incorporates a novel type of preprocessing capable of addressing the data imbalance. Such modifications produce a large positive impact on the quality of the results.

C3 **BF-PSR framework:** The BF-PSR framework is introduced as an extension of the PSR framework in such a way that we adequately insert our proposed behavioral features in a vector space, which remains non-sparse and low-dimensional as it is desirable, but now contains more valuable patterns that highlight the differences between groomer and non-groomer conversations. To achieve this contribution, the PSR* profiles and the seven proposed BFs are stacked together generating the BF-PSR vector representation as our formal proposal.

C4 **Assembly of two new datasets:** To mitigate the shortage of data in the area of online grooming are assembled and studied two new datasets, which we named as PJZ and PJZC. The two sets are publicly available for download aimed at fostering further researches in this area.

C5 **Benchmark experiments:** To demonstrate that BFs truly contribute to the early detection of online grooming, our proposal is compared against the state-of-the-art methods of the preventive scenario, i.e., PSR (ESCALANTE *et al.*, 2017), ENB (ESCALANTE *et al.*, 2016) and the three variants of method MulR (LÓPEZ-MONROY *et al.*, 2018). The results reveal that by employing the BF-PSR framework in the complete SGD testing set as well as in the new proposed datasets, i.e., PJZ and PJZC, we obtain state-of-the-art performance in the area of early grooming detection. Besides, because the proposed BFs and the improved PSR* vector obtained discriminatory patterns for each class, our method performs well despite having a large amount of unbalanced data, and therefore, it was not necessary to use any filtering technique to obtain highly accurate results.

Figure 1 summarizes some of the results obtained from the complete SGD dataset. It reports the accuracy of results achieved with different percentages of information available so to simulate the sequential arrival of online messages. As it can be seen, our proposed framework BF-PSR outperformed the four methods from the state-of-the-art in the area of early detection of grooming through **every single** portion of available information. These results also demonstrate how existing methods do not perform well with the full SGD corpus. Through this monograph, we provide details on how these state-of-the-art results were obtained.

1.1 Organization

The remainder of this monograph is organized as follows. The theoretical foundation is described in Chapter 2. Then, due to the importance of the SGD corpus for preventive scenario tools, a detailed description of its properties is given in Chapter 3. The related works are presented in Chapter 4. In Chapter 5, the seven behavioral features that we propose are described and analyzed. Chapter 6 introduces our framework BF-PSR. Chapter 7 is devoted to results and discussions. Chapter 8 concludes the monograph.

Besides the aforementioned content, in Appendix A, we propose the implementation of a triggering mechanism to be used in combination with our BF-PSR framework. Such mechanism decides “when” the classifier should output a reliable prediction with as little text as possible to give an alert. In this monograph, alerts must be produced when there is suspicion of grooming attacks in online conversations. Also, to the best of our knowledge, this work is the first one to employ a triggering mechanism in the field of early grooming detection.

CHAPTER 2

THEORETICAL FOUNDATION

This chapter presents several concepts that make up the theoretical foundations of text classification. Mostly, text classification is composed of traditional steps such as preprocessing and vectorization of the text. Then, a classification algorithm is chosen to process and assign a class to the data, and finally, evaluation metrics are applied to assess the effectiveness of the model. In summary, the main objective is to assign a category to a document by evaluating its components (CRIJNS, 2016). Each of these steps is described in the following.

2.1 Data preparation

Because the messages of a conversation are unstructured data, i.e., unorganized data with a not predetermined format, it is necessary to transform this data into its numerical format to obtain relevant information. To achieve this goal, the following tasks are performed.

2.1.1 Text preprocessing

To obtain relevant information from noisy and unstructured text documents we can start by preprocessing its content. Text preprocessing allows machine learning models to get clean and structured information from text (KADAM, 2020). Besides, depending on which techniques are applied, there may be a high or low impact on the system performance. The following is a description of the preprocessing techniques commonly employed in the literature. Nevertheless, it should be understood that no rule dictates that all preprocessing techniques must be applied. Which techniques to apply and in what order depend on both the dataset and the classification task of interest. Further details on the following techniques are found in Naseem, Razzak and Eklund (2020), Sarkar (2019).

Removing XML attributes: Most datasets are compressed in XML files. So often the first step is to extract the conversations from these files and remove the XML attributes such as tags or entities that are embedded in the data and add noise to the information. Table 1 presents an example of this technique.

Table 1 – Removing XML attributes preprocessing technique.

Before preprocessing	**After preprocessing**
<conversation id=eff3432hg> hii! my name is Jake, asl? </conversation>	hii! my name is Jake, asl?

Removing or replacing URLs: chat conversations usually contain URLs/links that, in most cases, do not provide relevant information. So a common practice in text preprocessing is to remove all URLs or replace them with a specific token such as <url_token>. Table 2 presents an example of this technique. In this specific case, the URL https://tinyurl.com/y4cm2b3q was replaced by the specific token <url_token>.

Table 2 – Removing or replacing URLs preprocessing technique. Based on (NASEEM; RAZZAK; EKLUND, 2020).

Before preprocessing	**After preprocessing**
This is an illustration of #theartoftweeting for the benefit of @scottmorrison https://tinyurl.com/y4cm2b3q	This is an illustration of #theartoftweeting for the benefit of @scottmorrison <url_token>

Removing or replacing punctuation, numbers, and special characters: Punctuation removal, numbers removal, and special characters removal form part of the classic text preprocessing techniques. They consist of removing those non-alpha symbols that do not provide relevant information and introduce noise to the text such as (!?.;,:-&$%12345). As with the URLs, this characters can also be replaced with specific tokens such as <number_token>, <puntuation_token>, etc. Table 3 presents an example of this technique. Here, both punctuation and numbers characters were replaced with specific tokens.

Table 3 – Removing or replacing punctuation, numbers, and special characters preprocessing techniques. Based on (NASEEM; RAZZAK; EKLUND, 2020).

Before preprocessing	**After preprocessing**
hey so many time changes for UA 1534. We going tonight or what? Missing In Action :(	hey so many time changes for UA <number_token>. We going tonight or what <puntuation_token> Missing In Action :(

Lowercasing tokens: It is the process of converting tokens from uppercase to lowercase. This technique simplifies the process of matching a term in the corpus, i.e., “internet” and

“INTErNET” are the same tokens if they are lowercase. Also, with this technique we can reduce the size of the vocabulary considerably. However, if the classification task consists of identifying entities as proper names, e.g., names of people or cities, this technique should not be applied. Table 4 presents an example of this technique. From the example it can be observed that the entities “Brat Pitt” and “LA” were lost due to preprocessing.

Table 4 – Lowercasing tokens preprocessing technique.

Before preprocessing	After preprocessing
`i saw Brat Pitt yesterday in LA #hollywoodlife`	i saw brat pitt yesterday in la #hollywoodlife

Removing stopwords: It is the process of removing words that have little or no significance. Words like “a”, “the”, “is”, “and”, “am”, “are”, “on”, etc., are usually words that end up having the maximum frequency in a corpus. This list of words is commonly referred to as stopwords. Table 5 presents an example of this technique. As it can be observed, the stopwords “was”, “until”, “I”, “the”, “of” were removed from the post.

Table 5 – Removing stopwords preprocessing technique. Based on (NASEEM; RAZZAK; EKLUND, 2020).

Before preprocessing	After preprocessing
`I thought Comcast was bad, until I saw the bad side of United Airlines`	thought Comcast bad, saw bad side United Airlines

Stemming and lemmatization: Stemming/lemmatization is the process of grouping together the inflected forms of a word so they can be analyzed as a single item identified by the word’s stem/lemma (KETTUNEN; KUNTTU; JäRVELIN, 2005). For example, the steam of “studies” is “studi” and the lemma of “studies” is “study”. Lemmatization, unlike stemming, ensures that the root word belongs to the language. Table 6 presents an example of this technique. Here the words “bats” and “feet” were lemmatized to their inflected forms “bat” and “foot”, respectively .

Table 6 – Lemmatization preprocessing technique.

Before preprocessing	After preprocessing
`The striped bats are hanging on their feet for best`	The striped bat are hanging on their foot for best

Replacing abbreviations and slang: Abbreviations and slang terms are commonly found in informal texts such as social networks posts or opinion platforms where no technical or formal vocabulary is required to engage in a conversation. Thus, the authors of the conversation employ colloquial terms that make harder text preprocessing. An example of it can be found in the term "asl" which is the abbreviation of the phrase "age, sex, location". The proper solution is to replace the term with its formal meaning so to avoid loosing relevant information. Many repositories on the Internet collect and make available the meanings of the most common abbreviations and slang terms. Table 7 presents an example of this technique. It can be seen how the abbreviation "asl" was replaced by its formal meaning.

Table 7 – Replacing abbreviations and slang preprocessing technique.

Before preprocessing	**After preprocessing**
`hiii asl, are you female or male?`	hiii age,sex,location, are you female or male?

Removing or replacing emoticons and emojis: Emoticons and emojis are digital characters frequently used to express feelings or opinions. As with the other techniques, depending on the task, we can choose to remove or replace the emoticons or emojis by their meaning or by a specific token. For example, the sequence of characters `:-*` can be removed, replaced by its meaning "kisses" or it can be replaced by a specific token `<emoticon_token>`. Table 8 presents an example of this technique. In this case, the emoticons `:-*` and `:)` were replaced by their corresponding meanings.

Table 8 – Removing or replacing emoticons preprocessing technique.

Before preprocessing	**After preprocessing**
`gr8! sending tons of :-* right now :)`	gr8! sending tons of kisses right now happy

Word segmentation: Social conversations are replete with hashtags that can contain relevant information. Thus, word segmentation is the process of separating phrases compressed into a hashtag to retrieve valuable information. For example, `#peoplerightsmatter` is segmented as three tokens: `people`, `rights` and `matter`. Table 9 presents an example of this technique. Here the hashtag `#goodvibes` was segmented into the tokens `good` and `vibes`.

Table 9 – Word segmentation preprocessing technique. Based on (NASEEM; RAZZAK; EKLUND, 2020).

Before preprocessing	**After preprocessing**
`#goodvibes United Airlines Flies Children With Serious Illnesses To Santa North Pole`	good vibes United Airlines Flies Children With Serious Illnesses To Santa North Pole

There are many other methods for preprocessing text such as to replace elongated characters, to correct misspelling words, to expand contractions, etc. They are not detailed in this monograph for brevity.

2.1.1.1 Discussion

As mentioned at the beginning of this subsection, depending on which technique or combination of techniques is applied there may be a variation in the performance of the classification model. For example, although one of the mentioned techniques was to remove emoticons, whether or not to perform this process depends mostly on the classification task and on the dataset to be used. Removing the emoticons could have a negative impact if the documents of the data are scarce or if the task is oriented to sentiment analysis. Additionally, the order in which the techniques are applied may also affect the performance of the classification model. If in the sentence "i am feeling pretty gr8 now" we first remove the numbers, the term `gr8` is lost; and, as a consequence, so it is the meaning of the abbreviation, i.e., the word "great". One must always keep in mind that the goal is to maintain valuable information, not to lose it.

The following subsection describes how to convert text into a format that machine learning models can handle.

2.1.2 Text representation

Machine learning algorithms usually expect numbers as input, so it is necessary to transform the dataset into its numerical format (KADAM, 2020). To achieve this objective the text representation task is divided into two main steps: tokenization and vectorization.

Tokenization and n-grams: The goal is to compose a vocabulary with the unique tokens of the whole corpus through text tokenization. In this way, the tokenization step refers to how the text is split into tokens. There are several types of tokenization. For example, if the emoticons are not important for the classification task, the tokenization can be performed by punctuation. Let us consider the following post[1]:

```
CANT WAIT for the new season of #TwinPeaks \(^o^)/ yaaaay!!!
#davidlynch #tvseries
```

If we tokenize the post by punctuation the result is as follows:

```
["CANT", "WAIT", "for", "the", "new", "season", "of", "#",
"TwinPeaks", "\(^","o", "(^/","yaaaay","!!!", "#", "davidlynch", "#",
"tvseries"]
```

[1] https://github.com/cbaziotis/ekphrasis

As it can be observed, for this particular case, the composition of the emoticon is lost and it is not clear how the other punctuation tokens are split. On the contrary, if the emoticons are important for the task, a different type of tokenization can be applied to preserve the emoticons:

```
["CANT", "WAIT", "for","the", "new", "season", "of", "#TwinPeaks",
"\(^o^)/", "yaaaay", "!", "!"," !","#davidlynch", "#tvseries"]
```

Here, we not only preserve the emoticon but also the hashtags. The first style of tokenization is referred to as punctuation tokenizer and the second as social tokenizer. To know which of the different tokenizers to apply, it is necessary to understand the goal of the classification task. Once the tokenization is performed, all the unique tokens must be saved into a vocabulary.

To alleviate the loss of context, it is possible to use n-grams to represent the vocabulary text. The process consists of using n-consecutive tokens, e.g., words, characters, sentences, to represent the text. In this way, partial information about the order of the words (or characters, or sentences) is maintained (ROA, 2018). For example, for a 2-gram representation considering words as tokens, the sentence “Mary Jane send :-*”, is split into “Mary Jane”, “Jane send”, “send :-*”. This means that the name "Mary Jane" will be taken into account as a whole in the first 2-gram, instead of having individual tokens per word, i.e., "Mary","Jane". Therefore, depending on the choice of n-grams, the final vocabulary can be composed of 1-grams, 2-grams, n-grams, or all of them.

Vectorization The step of vectorization is the one that transforms the tokens into numerical vectors (KADAM, 2020). Next, we described some of these techniques. Furthermore, a small corpus composed of two posts already preprocessed and tokenized is depicted in the following for a better exemplification of this process.

Corpus example

```
Post 1: ["the","rights","of","one","are","the","rights","of","all"]
Post 2: ["for","humanity","for","the","change"]
```

i) One-hot encoding: For each token of the vocabulary a unique index is assigned. Then each document is represented as a vector indicating the presence, i.e., 1 or absence, i.e., 0 of a token in the text. An example of this representation is depicted in Table 10 where documents Post 1 and Post 2 are represented as one-hot vectors. Besides, when all the documents of the corpus are represented in their vector format, the whole structure is named as term-document matrix, where the rows are the unique terms, i.e., the vocabulary of the corpus, and the columns are the documents/posts.

ii) Count encoding: In the case of count encoding, the vector indicates the frequency of the

Table 10 – One-hot encoding vectorization example.

Vocabulary	Post 1	Post 2
the	1	1
rights	1	0
of	1	0
one	1	0
are	1	0
all	1	0
for	0	1
humanity	0	1
change	0	1

tokens in the document. An example of this representation is depicted in Table 11, where documents `Post 1` and `Post 2` are represented through a count vector encoding. Here, instead of having binary values in the term-document matrix cells, we have the frequency of the token in the document.

Table 11 – Count encoding vectorization example.

Vocabulary	Post 1	Post 2
the	2	1
rights	2	0
of	2	0
one	1	0
are	1	0
all	1	0
for	0	2
humanity	0	1
change	0	1

iii) TF.IDF: Although one-hot vector encoding and count encoding are common vector representations, the literature confirms that knowing if a token is present or not in a document is not enough to achieve satisfactory results with machine learning methods. A solution is to assign weights to the tokens aimed at representing their importance in the document. The algorithm Term Frequency Inverse Document Frequency (TF.IDF) achieves this goal. It combines two metrics, Term Frequency (TF) and Inverse Document Frequency (IDF) which is mathematically represented by the product of these metrics, as it is shown in Equation 2.1.

$$\text{TF.IDF} = \text{TF} \cdot \text{IDF} \tag{2.1}$$

The TF value can be calculated through Equation 2.2, where f_{w,d_i} denotes the frequency f of word w in the i^{th} document of the corpus.

$$\text{TF}(w, d_i) = f_{w,d_i} \tag{2.2}$$

Them, the IDF value is represented by Equation 2.3. Here, N represents the total number of documents in the collection, and $Df(w)$ represents the number of documents D in which the word/term w appears. The log function is later applied to soften the result from dividing N by $Df(w)$. As result, the IDF estimates if the word is common or rare in the document collection (SARKAR, 2019). Thus, the TF value increases proportionally to the number of times a word appears in the document, but it is compensated by the word frequency in the document collection. The resulting TF.IDF value is later placed in the term-document matrix instead of the binary values of the one-hot vector representation.

$$\mathrm{IDF}(w,D) = 1 + \log(\frac{N}{1+Df(w)}) \tag{2.3}$$

2.1.2.1 Discussion

Representing textual data in its numerical format presents some difficulties. For example, many of the documents in the corpus may not contain several of the vocabulary tokens, so the term-document matrix becomes sparse which causes that a large amount of memory to be misused. In addition, the more documents and unique tokens, the slower the processing of the classification model. Another issue is the fact that none of the vector representations preserve the meaning of the tokens. A solution for this latter disadvantage is to employ word embeddings that do take into account the semantic meaning of the words. However, this type of representation suffers some drawbacks such as high dimensionality and sparsity.

In the next section, we describe the main machine learning classifiers oriented to text processing.

2.2 Machine learning binary classifiers

After preprocessing and representing the textual documents into their numerical format, we can employ Machine Learning (ML) algorithms to classify or categorize the data. Next, we briefly describe the most popular ML algorithms used for text classification. Provided that the focus of our work is the detection of grooming, we limit the description to binary classification, thus considering that our problem has only two classes, groomer and non-groomer. In this sense, multi-class classification metrics are out of the scope of our work. For a more detailed explanation of these algorithms refer to (SARKAR, 2019; KADAM, 2020).

Naïve Bayes: The Naïve Bayes (NB) classifier is a supervised learning algorithm, i.e., the labels of the dataset are known and the algorithm learns to predict them, based on the very popular Bayes' theorem. The classifier aims to find the class $\widehat{C}$ from a set of n classes $C = \{C_1, C_2, ..., C_n\}$ with the maximum probability given a text document A. Also, because we are working with binary classifiers the value of $n = 2$. This process is mathematically described in Equation 2.4:

$$\widehat{C} = arg\ max_i\ P(C_i \mid A) \tag{2.4}$$

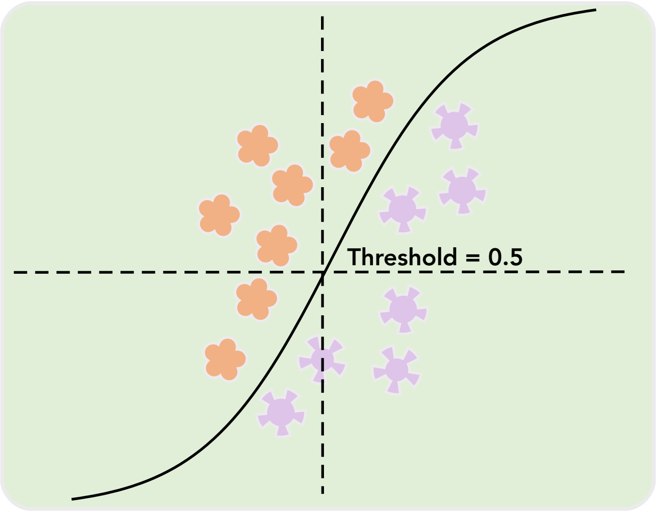
To find the most probable class $\widehat{C}$, the posterior probability is calculated using the Bayes' theorem depicted in Equation 2.5. To find whether C_i is the most probable class for a given document A, one must first calculate the probability that A belongs to class C_i multiplied by the probability of class C_i and divide the result by the probability of the text document A.

$$P(C_i \mid A) = \frac{P(A \mid C_i)P(C_i)}{P(A)} \tag{2.5}$$

Later, it is possible to set a threshold value to determine whether the document A belongs to class C_i or not. For example, if the output probability of the classifier is higher than 0.6, it belongs to class C_i (CRIJNS, 2016). Finally, because this classifier considers all features to be independent, it receives the name of "naïve".

Logistic regression: The Logistic Regression (LR) classifier uses the sigmoid mathematical function to estimate the text document category. This function can be depicted mathematically by the following formula: $\frac{1}{1+e^{-x}}$, where e is the Euler exponent and x depicts the text document in its mathematical representation. The model is illustrated in Figure 2 where, for this specific case, the assigned threshold value is 0.5. Therefore, the LR model returns 1 if the threshold is equal or greater than 0.5 and returns 0 otherwise.

Figure 2 – Logistic Regression as text classification model with threshold value equals to 0.5.

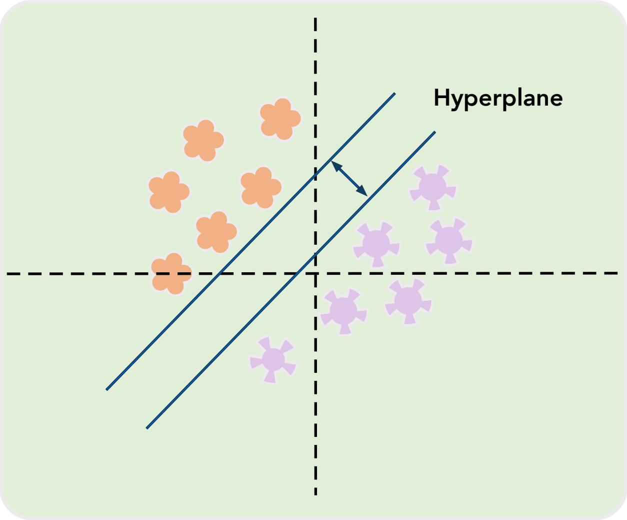


Source: Elaborated by the author.

Support vector machines: Popularly know as SVMs, Support Vector Machines are supervised learning algorithms. The classifier aims to represent the training data documents as points in the space. Then, it learns to separate the points belonging to either class through a hyperplane, and finally, the new data points to be predicted are assigned to classes based on which side of this hyperplane they fall into (SARKAR, 2019). Figure 3 illustrates this process.

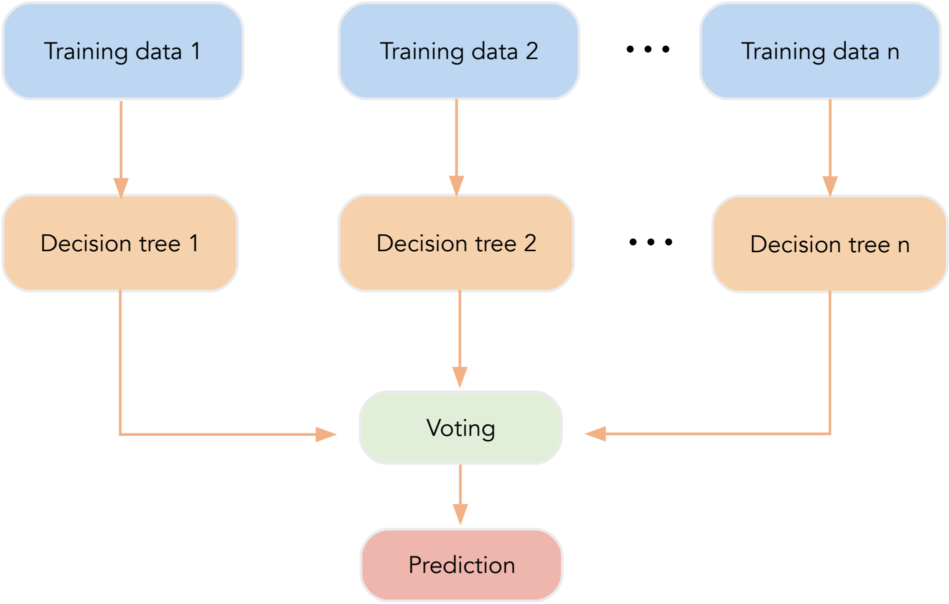
Random forest: The Random Forest (RF) classifier is based on the concept of ensemble learning, which is the process of combining multiple classifiers to solve a complex problem

Figure 3 – Support Vector Machine as text classification model. The model learns to generate a hyperplane that separates the two classes. Based on (SARKAR, 2019).



(JAISWAL, 2021). In this way, the RF classifier uses multiple decision trees built for subsets of documents selected at random from the database to perform the classification. More specifically, the decision tree splits the dataset into subsets while simultaneously building more decision trees. The process occurs in parallel. Next, to obtain the output, RF takes the prediction from each tree and based on the majority votes of predictions it predicts the final output. Furthermore, a greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. A representation of this process can be observed in Figure 4.

Figure 4 – Random Forest as text classification model. Subtrees are created in parallel to vote for the final answer. Based on (JAISWAL, 2021).



Multilayer perceptron: A Multilayer Perceptron (MLP) is a type of neural network that can be described as a set of layers, i.e., input layer, hidden layer and output layer. These models are also known as Feed Forward Networks (FFN) because the information flows from the input (x), through the hidden layers used to define a function (f), to the output (y). Under this concept, a MLP is simply a mathematical function mapping some set of input values (x) to a category (y) and on the way, they learn the value of the parameter weights (θ), which results in the best function approximation (GOODFELLOW; BENGIO; COURVILLE, 2016). Mathematically, this mapping process is given in Equation 2.6.

$$y = f(x, \theta) \tag{2.6}$$

Regarding its architecture, the model is composed of three fundamental layers:

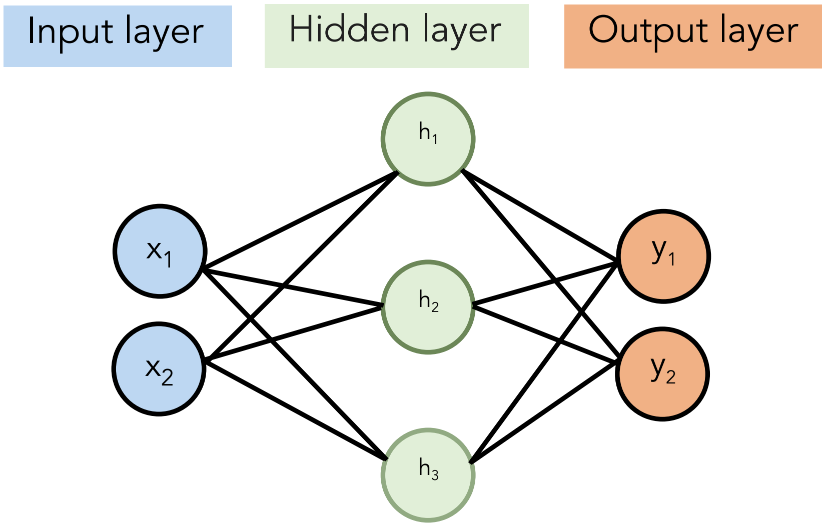
i) Input layer: it receives the features of the dataset.

ii) Hidden layers: the hidden/intermediate layers (h) compose the function of the network.

iii) Output layer: it is the last layer of the network and conserves the final result.

Figure 5 shows the basic structure of a MLP with two inputs, one hidden layer with three nodes, and two outputs. The output chosen will be the one with the highest probability.

Figure 5 – Basic structure of a Multilayer Perceptron . Based on (GRON, 2017).



In addition, to understand the great popularity of MLPs, it is necessary to introduce the concept of "backpropagation". Backpropagation is a process that repeatedly adjusts the weights (θ) of the connection layers to decrease the error in the predictions (RUMELHART; HINTON; WILLIAMS, 1986). So, it is basically an optimization problem whose goal is to minimize the error between the network output (y_i) and the true label. One can refer to this iterative process as "learning". The learning process is divided into two main phases:

i) Forward phase: we already described this phase to understand the FFNs. Briefly, the input x_i is propagated, layer by layer, through the network until it produces an output y_i. Refer Equation 2.6.

ii) Backward phase: after performing the forward phase, an error is calculated by comparing the network output with the true label. Then, this error is propagated in a backward direction. In each iteration, the weights are adjusted until we arrive at the first layer where the weights θ are updated and the process starts again until the error value gets small. The update process is performed with respect to a cost function. The most famous cost function is commonly known as the Stochastic Gradient Descent (SGD) cost function (KIEFER; WOLFOWITZ, 1952), which over the years has been improved by several variants such as the Adaptive Gradient (AdaGrad) algorithm (DUCHI; HAZAN; SINGER, 2011), the Root Mean Square Propagation (RMSProp) algorithm (HINTON, 2020), the Adaptive Moment Estimation (Adam) algorithm(KINGMA; BA, 2015), etc.

To conclude this topic, it is important to know that there are a number of other models used to classify text, e.g., k-Nearest Neighbor (MUCHERINO; PAPAJORGJI; PARDALOS, 2009), Decision Trees (QUINLAN, 1986), Long Short-Term Memory (HOCHREITER; SCHMIDHUBER, 1997), etc. However, we consider that those described above are sufficient to understand the theoretical foundations of this monograph.

2.3 Evaluation and metrics

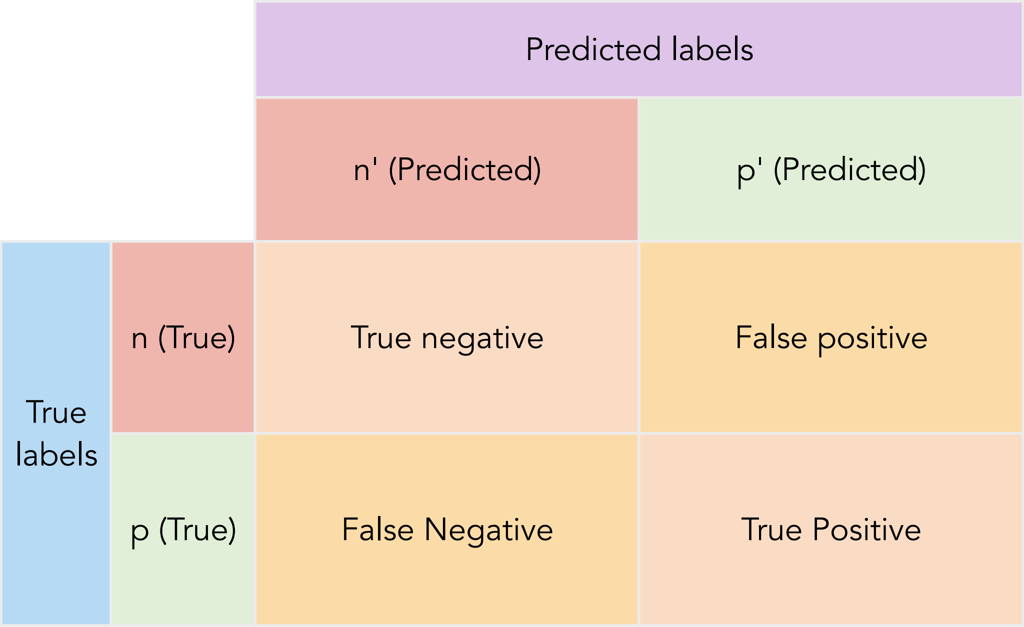
Once we have the classifier trained, it is necessary to evaluate its performance. Below, we mention some of the most important metrics to evaluate a classifier, such as Confusion matrix, Accuracy, Precision, Recall, and F1$_g$ metrics. Note that this section only describes binary classification metrics, since it is the type of classification required by our target application.

Confusion matrix Although it is not a formal evaluation metric, the confusion matrix keeps the number of correct and incorrect classified instances in a table structure. Various evaluation metrics are derived from this information (SARKAR, 2019). In Figure 6, we can see its structure in more detail. Where n and p are the negative and positive classes respectively. The True Negative (TN) cell contains the number of instances correctly classified as negative; the True Positive (TP) cell contains the number of instances correctly classified as positive; the False Positive (FP) cell contains the number of instances incorrectly classified as positive, and; the False Negative (FN) cell contains the number of instances incorrectly classified as negative. As a simple observation, the classifier's objective is to maximize the number of instances in the true cells of the matrix. That is, the diagonal of the matrix.

In addition, it is important to note that in a given dataset, the positive class is the one we are trying to detect. Thus, for this specific work, we refer to the positive class as the groomer conversations and the negative class as the non-groomer conversations.

Accuracy metric Accuracy is defined as the overall proportion of correct predictions of the model. Equation 2.7 presents its formulation. Accuracy metric works well when the data is

Figure 6 – Typical confusion matrix structure. Based on (SARKAR, 2019).



		Predicted labels	
		n' (Predicted)	p' (Predicted)
True labels	n (True)	True negative	False positive
	p (True)	False Negative	True Positive

balanced and the correct predictions of the classes are equally important. Otherwise, this is not a reliable metric (GRON, 2017).

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{2.7}$$

Precision metric The Precision metric is only focused on the correct values of the positive class. Equation 2.8 presents its formulation. It becomes important when we need to find the maximum number of positive classes even if the total Accuracy reduces (SARKAR, 2019).

$$Precision = \frac{TP}{TP+FP} \tag{2.8}$$

Recall metric Also called *sensitivity*, it is the ratio of positive instances that are correctly detected by the model. Equation 2.9 presents its formulation. Recall becomes an important metric when we need to catch the maximum number of instances of a particular class even when it increases our false positives (SARKAR, 2019).

$$Recall = \frac{TP}{TP+FN} \tag{2.9}$$

F1$_g$ measure Its original notation is F1 measure or F1-score, however, because our objective is to detect the positive class, i.e., the groomer conversations, in this work we decided to refer to this metric as F1$_g$ to emphasize that we are interested in calculating the errors when detecting the groomer conversations. Having clarified this, F1$_g$ helps to optimize the classifier for balanced Precision and Recall performance (GRON, 2017).

$$\text{F1}_g \text{ measure} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.10}$$

2.4 Summary and discussion

In this chapter, we have briefly introduced several concepts that make up the theoretical foundations of text classification such as data preprocessing, text representation, machine learning algorithms, and evaluation metrics. We believe that these concepts are sufficient to understand the theoretical basis of our proposal. In the next chapter, we will describe in detail the SGD dataset.

CHAPTER 3

SEXUAL GROOMING DETECTION DATASET

As it was mentioned in the introductory Chapter 1, the conversation between a child and a groomer may contain sensitive information that cannot be published on the Internet, so freely available datasets are very rare in the area of online grooming detection. Thus, we have decided to describe the Sexual Grooming Detection (SGD) dataset[1] in this chapter, which to the best of our knowledge is the most popular and complete dataset available publicly in the context of grooming detection. Furthermore, due to its intrinsic characteristics, it has become the only dataset employed by tools performing in a preventive scenario. All these reasons turn the SGD dataset into a valuable resource for the area of grooming detection. Besides, we believe that by detailing this corpus one can better understand the related works summarized in Chapter 4, as well as our proposal, that also makes use of this corpus to make a fair comparison with the concurrent methods in the area of early grooming detection.

Dataset SGD was first used in the international sexual groomer identification competition at PAN[2] 2012. The objectives of the contest were twofold: a) to identify the groomers among all users in the different conversations, and; b) to identify the lines of the conversations that are the most distinctive of groomer behavior. To evaluate the performance of the participants, the authors of (INCHES; CRESTANI, 2012) created a corpus with a long collection of conversations that attempt to mimic the properties of a realistic scenario. Thus, the authors decided that the groomer class of the corpus should be represented by less than 4% of the conversations in the whole dataset since in real-life cases, conversations with this content are scarce compared with regular conversations. In this sense, there are three sources of conversations that compose the collection:

i) True Positives (TPs): logs of online conversations between convicted sexual groomers

[1] Note that the original name of the dataset that we use is Sexual Predator Detection (SPD) dataset. However, to maintain the uniformity in our terminology, we refer to it as the Sexual Groomer Detection (SGD) dataset.

[2] <https://pan.webis.de>

and volunteers posing as underage teenagers. The chats were obtained from the American Perverted Justice Foundation (PJ) website[3].

ii) False Positives (FPs): chats containing abusive language, general silliness, and cybersex. They were extracted from Omegle[4].

iii) True Negatives (TNs): regular conversations that were aggregated to the collection to add variety to the topics and increase the number of interactions. The chats were extracted from two IRC logs[5,6].

Because the conversations were obtained from different sources, the authors only kept those chats with a maximum duration of 25 minutes and with no more than 150 messages, thus, the different files could be comparable.

3.1 Characteristics of the corpus

The corpus is originally divided into training and testing sets, stored in XML files with a structure of logical trees as it is depicted in Figure 7. The root of the tree is called "conversations". The root has many children called "conversation", each one with its identification. In addition, each "conversation" is divided into multiple messages labeled with line numbers. Then, each "message" has the following attributes: "author" of the message, i.e., the value of a randomly generated identifier for the author, "time" when the message was sent (HH:MM), and the content of the message under the "text" tag. Besides, each XML file is accompanied by a list of authors cataloged as groomers. Thus, after properly extracting the information from the XML files, it was possible to obtain basic metadata from the corpus. As it is shown in Table 12, the total number of conversations in the training set is 66,927. They come from 97,689 unique users and only 142 of the unique users are groomers. For the testing set, the number of chats is 155,128, which is nearly twice the number of training conversations. There are 218,702 unique users, and 254 of them are groomers.

Table 12 – Number of conversations, users and groomers in the original training and testing sets.

	Training dataset	**Testing dataset**
Conversations	66,927	155,128
Unique users	97,689	218,702
Unique groomers	142	254

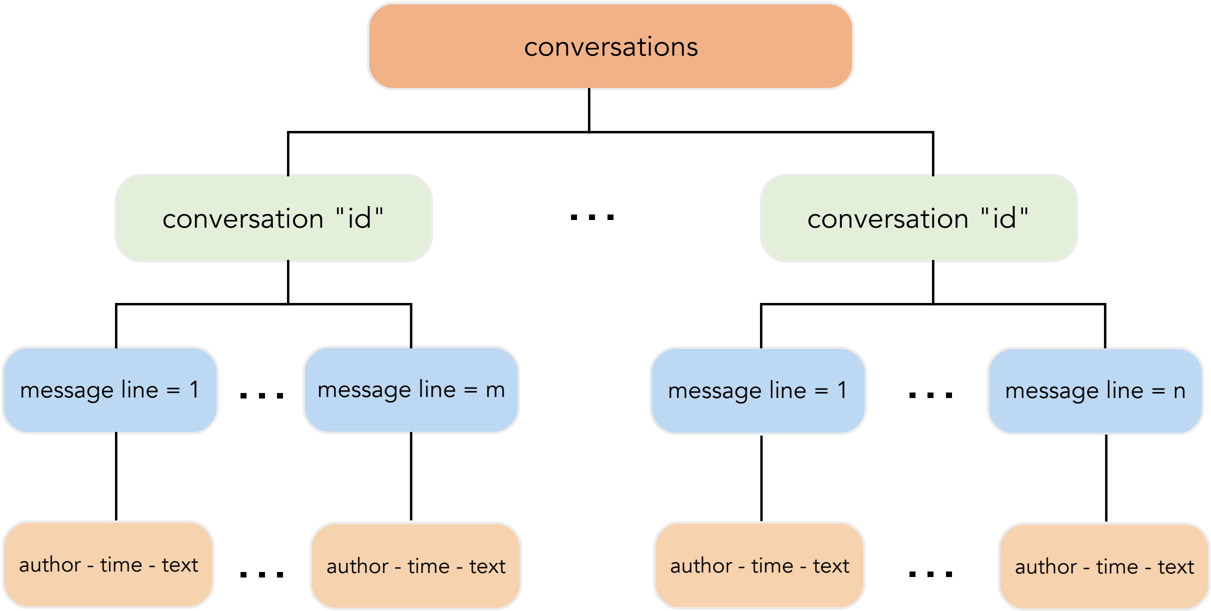
[3] <http://www.perverted-justice.com>

[4] <http://omegle.inportb.com/>

[5] <http://www.irclog.org/>

[6] <http://krijnhoetmer.nl/irc-logs/>

Figure 7 – Logical tree structure of the SGD dataset. Based on (BOURS; KULSRUD, 2019).



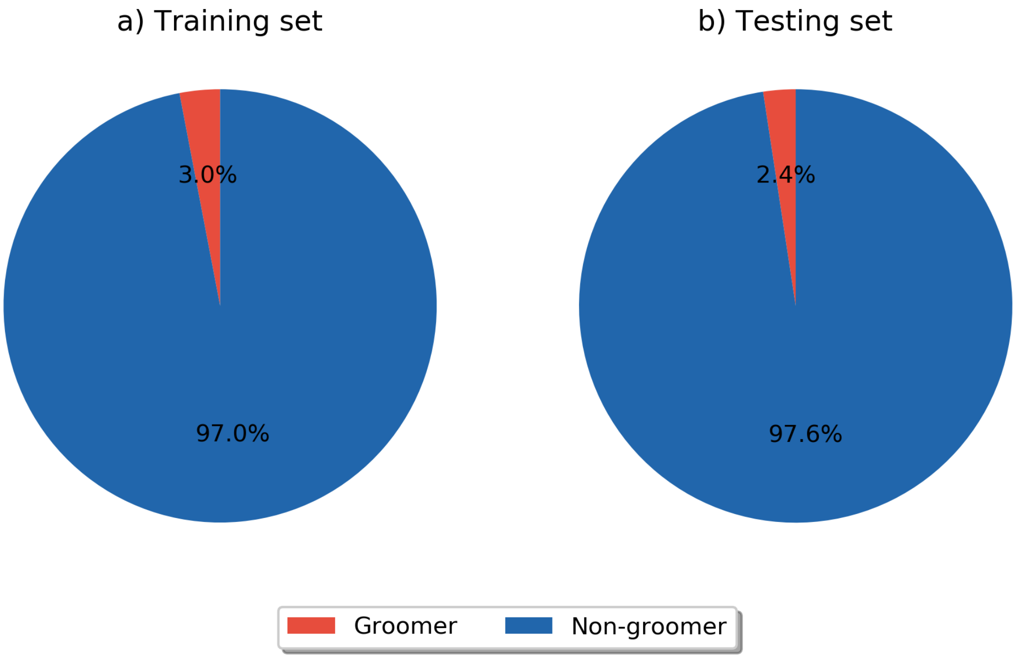
3.2 Objective and evaluation on tasks of early detection

From the investigation of the SGD corpus, we found out that this dataset is also employed for tasks of early detection. The objective is no longer to identify all groomer authors in a set of conversations as in the original competition, but rather, to identify a suspicious conversation as early as possible. To make it possible, some adjustments were made to the original structure of the dataset because it is not oriented to this specific task. Thus, a relabeling process was carried on in a similar way to the one performed in VILLATORO-TELLO *et al.* where each conversation that contains at least one groomer as a participant was considered to be a suspicious conversation and labeled with the groomer (G) class; the remaining conversations were labeled with the non-groomer (NG) class. Table 13 reports the characteristics of the corpus when oriented to the early detection of groomers. Note that there are empty conversations in the corpus; they are ignored in our work. Therefore, there are 2,015 G-conversations in the training set and 3,724 ones in the testing set. Also, there are 64,899 NG-conversations in the training set and 151,377 ones in the testing set. Figure 8 highlights the extreme unbalance of the SGD corpus; note that the G-conversations represent less than 4% of the whole dataset. Finally, because the objective is to detect the groomer conversations, the F1 measure of the positive class (depicted in Chapter 2) is commonly used in the literature to evaluate early detection tasks. In this work, we refer to this metric as $\mathrm{F1}_g$, where g refers to the groomer class.

Table 13 – Number of groomer and non-groomer conversations available for the early detection task. Note that the numbers under parentheses report the empty conversations that were removed from the training and the testing sets.

	Training dataset	Testing dataset
Groomer conversations	2,016 (-1)	3,737 (-13)
Non-groomer conversations	64,911 (-12)	151,391(-14)

Figure 8 – Percentage of groomer and non-groomer conversations in the training and the testing sets. In both sets, the G-conversations represent less than 4% of the whole SGD data.



3.3 Summary and discussion

In this chapter, the most popular dataset regarding the detection of online grooming was outlined. The SGD corpus is employed in most of the related works and also in our work. From the characteristics of the set, it was observed that the data is extremely unbalanced because the authors of the corpus intended to create a collection that reflects a real scenario, where there are regular chats (TN), groomer chats (TP), and consensual sex chats between adults (FP). In addition, we described how the corpus was adapted to be used in tasks of early detection of grooming and highlighted that the most indicated metric to evaluate the performance of preventive methods is the $F1_g$ score, i.e., F1 measure of the positive, groomer class.

CHAPTER 4

RELATED WORKS

Online grooming has become a serious problem due to the huge amount of online applications that children and young people interact with. In these messaging platforms, the groomer may pose as a child and establish a relationship with the kid. In fact, most groomers follow a sort of "discourse model" which in the study of Lorenzo-Dus and Kinzel (2019) is described as a set of phases to approach a victim: a) **gaining access** – involves exchanging personal information through social media, video games, streaming apps, and the like; b) **deceptive trust development** – refers to the ability of the groomer to create a trusting relationship through compliments, common interests, or talks about feelings; c) **compliance testing** – it is the process by which groomers check whether their target is underage; d) **sexual gratification** – at this point, the groomer begins to display sexual behavior and manipulates the victim to do it as well; e) **isolation** – the groomer makes sure that the victim does not talk about their relationship with his/her family or friends, and; f) **approach** – the groomer requests to meet the victim offline with the intent of beginning a sexual relationship.

In an effort to prevent online grooming attacks, institutions such as the American Perverted Justice Foundation[1] were created to confront criminals with grooming behavior on the Internet. Unfortunately, despite its work since 2003, the foundation ceased to operate in 2019 due to the enormous amount of chats that must be monitored across multiple platforms. Similarly, several institutions suffer from the same difficulty, as a limited group of staff is obviously not enough to monitor all online conversations on the Internet. For this reason, various tools have been proposed aimed at the automatic detection of grooming attacks, both considering the forensic and the preventive scenarios. The following sections present an extensive review of the proposed methods in both scenarios.

[1] <http://www.perverted-justice.com>

4.1 Forensic scenario

Through the years, many studies have concentrated efforts on the development of tools that detect groomer behavior. The first attempts to address this problem fall under a forensic scenario because these tools can only be used when the attack has already occurred. Furthermore, it can be observed that most works in this scenario try to identify whether or not the chats belong to the phases of the groomer speech model to thus detect suspicious behavior. It is important to emphasize that some of the proposals in this scenario employ behavioral features in their approaches, commonly with a focus on how one user employs a word and not on which words the user employs (MORRIS; HIRST, 2012). These features are not always explicit in the dataset, so a deep analysis may be needed to extract them. Next, we describe some of the most important works for the detection of grooming in a forensic scenario.

The work of Pendar (2007) is perhaps one of the first studies toward the construction of an automatic recognition system of online groomers. The authors collected a set of 701 chat conversations between groomers and victims (pseudo-victims) from the PJ website. Then, the authors performed a user-level document representation by dividing each line produced by the victim and the groomer into different documents. Specifically, the conversations were separated by the content produced by each user, so that there is one document per user. In the following, they applied feature extraction techniques, e.g., extracted bi-grams with stop words removal, and then employed a SVM model to perform user classification. In Edwards and Leatherman (2009), the communication strategies of a groomer in online platforms were studied. 288 chat logs from the PJ website were considered and the authors developed a rule-based system called "ChatCoder" that classifies each chat line into categories of the grooming discourse model, e.g., approach and isolation. To make it possible, they performed a chat-level representation, i.e., to represent each conversation as a document, and created a dictionary containing terms and phrases that are common to net culture and luring language. The authors also created a graphical interface that highlights, with a specific color, the terms recognized by the system as suspicious. A similar approach was addressed in McGhee *et al.* (2011), where the authors also considered a set of chats from the PJ website, applied a chat-level representation, and developed techniques for matching human hand-coding of groomer posts with the phases of the grooming speech model. Thus, the method labels the lines of each conversation in the following categories: gaining access, sexual gratification, approach, and lines containing none of the classes. Also, during the development of their method, some features were proposed for the detection of groomer behavior, such as the number of first person pronouns in a line (e.g., "I", "me"), the number of personal information nouns (e.g., "age", "pic"), the number of activities nouns (e.g., "movie", "favorite"), the number of approach nouns (e.g., "hotel", "car"), among others.

Later on, with the launch of the international sexual groomer identification competition in 2012[2], a series of papers were produced with a focus on identifying groomer users among the

[2] <https://pan.webis.de/clef12/pan12-web/sexual-predator-identification.html>

regular ones in the SGD dataset, which was described in Chapter 3. Thus, in Kontostathis *et al.* (2012) an extended version of the “ChatCoder” software was proposed to tackle the competition’s objectives. First, the authors separated the training set by user, creating one file per user, i.e., user-level representation. This allowed them to quickly and easily collect statistics at the user level. Then, they collected a set of fifteen behavioral features from the dataset that served as input for their machine learning model. Among these features, there are categories, such as the number of approach nouns, approach verbs, family nouns, first person pronouns in a line, relationship words, etc. Because the SGD corpus is extremely unbalanced, the authors applied sampling techniques and used the Weka data mining tool kit[3] to construct a decision tree that identified the suspicious users. In Morris and Hirst (2012), the authors also collected the lines that a user wrote in different conversations and represented each document at the user level. Then, to distinguish between groomer and non-groomer users, the authors extracted lexical features from the data, like BoW, uni-grams, and bi-grams with stop words removal, and proposed a set of behavioral features, e.g., number of messages, number of conversations, and a blacklist of 122 n-grams that automatically flag a message as risky. Later, a SVM model was employed to perform two classifications: to distinguish groomer users from non-groomers users and groomer users from victims.

In Parapar, Losada and Barreiro (2012), the dataset was also grouped by users and three types of features were extracted: a) lexical, like TF.IDF and uni-grams with long words removal and non-lemmatization; b) psycholinguistic, using the Linguistic Inquiry and Word Count (LIWC)[4] software with more than 80 categories, and; c) user-level features, which capture some global aspects related to the activity of the individuals in chat-rooms, e.g., average message size, number of characters in a message, number of users in the conversation and average time between messages. Then, a SVM classifier with L2-regularization and penalization in the error was trained with the features to detect the groomer users. In the work of Villatoro-Tello *et al.* (2012), the authors proposed a two-stage classification strategy for the detection of misbehaving users. The first stage aims to identify a conversation as suspicious by using a chat-level representation, and the second stage processes the suspicious conversations to distinguish the victim from the groomer by following a user-level representation. To make it possible, the authors split the data by conversations and, if one conversation contained a groomer user, the whole chat was labeled as a groomer conversation. Then, to overcome the unbalance in the data, a filtering process was applied where conversations with only one participant, or containing less than 6 interventions per user or long sequences of unrecognized characters (apparently images) were removed. No text preprocessing was performed to preserve the writing style of the users and a SVM classifier was trained to identify the suspicious conversations. For the second stage, the authors divided the conversations by interventions of the victim and the groomer; then, they applied a BoW or TF.IDF representation to perform a classification with a SVM model that

[3] <https://www.cs.waikato.ac.nz/ml/weka/>

[4] <https://liwc.wpengine.com>

identified the misbehaving users. It is worth mentioning that this method obtained the best results at the SGD competition in 2012.

After the competition, many researchers continued to propose tools for the detection of grooming in online conversations. For example, in Cheong *et al.* (2015) the authors sought to detect groomer behavior in real video game chats, specifically in the MovieStarPlanet chat game. In order to achieve their goal, they represented a document at the chat level and built a set of features, like sentiment-based features with emoticons and terms belonging to the AFINN-111 lexicon, and a blacklist containing forbidden terms in the game. However, because most of the groomers are aware of these terms, they commonly introduce noise into the words to avoid the detection. To address this problem, the authors generated various combinations of the terms in the list to transform the original term, e.g., sex, into modified ones, e.g., seeeex. Finally, the use of several machine learning models, e.g., Naïve Bayes (NB), MultiLayer Perceptron (MLP), and Logical Regression (LR), was proposed to perform the predictions. Further on, in Bogdanova, Rosso and Solorio (2014) the authors focused on distinguishing consensual sex conversations from groomer conversations. To make it possible, random chats were collected from the PJ website as TP and the FP were represented by cybersex chat logs[5] and chat-room conversations from the NPS[6] corpus. Then, the authors performed a high level feature extraction to recognize groomer patterns in chat-level documents. The proposed behavioral features were: a) sentiment-based features, like positive and negative; b) emotion-based features, such as joy, sadness, anger, surprise, disgust, and fear; c) neuroticism features, e.g., personal pronouns, reflexive pronouns, obligation pronouns and emoticons, and; d) some features borrowed from McGhee *et al.* (2011), like approach words, family words and information words. The use of deep learning was then proposed to detect online grooming in a forensic scenario. In Ebrahimi, Suen and Ormandjieva (2016), an architecture based on Convolutional Neural Networks (CNN) was proposed for the identification of groomer conversations with a chat-level representation. The authors employed the SGD corpus and studied the use of different features, like pretrained word embedding versus internal word embedding or Bag of Words (BoW) feature encoding versus one hot feature encoding. Then, they demonstrated that unlike in image classification, a deep CNN architecture was not beneficial for the text preprocessing, so a single layer was sufficient. The authors also found out that the removal of stop words and punctuation decreased the performance of their method.

Later on, inspired by various approaches proposed at the SGD competition, the authors of the work Cardei and Rebedea (2017) employed a two-stage classification strategy using both chat and user level representations to detect groomers. At first, they addressed the data imbalance by applying filtering techniques, i.e., only conversations with exactly two users and more than 20 messages were considered. Then, behavioral and interactional features were proposed by the authors, e.g., percentage of questions asked, underage expressions ratio, percentage of words

[5] <oocities.org/urgrl21f/>
[6] <http://faculty.nps.edu/cmartell/NPSChat.htm>

that are present in WordNet, percentage of messages that contain a negation, percentage of slang words, and the Flesch reading ease score. Their experiments indicated that by employing BFs for the detection of grooming they were capable of surpassing the winners Villatoro-Tello *et al.* (2012) of the competition PAN 2012. In Zuo *et al.* (2018), the authors constructed a subset data from the PAN 2013 author profiling dataset (RANGEL *et al.*, 2013) by selecting the classes pedophile and sex. They performed text feature extraction with BoW and TF.IDF at a chat-level representation, and then applied the fuzzy rough method for the feature selection and machine learning algorithms for binary classification, considering normal and abnormal documents, and also for multi-label classification, considering normal, pedophile, and sex documents. More recently, in Lykousas and Patsakis (2020) the authors collected public chats between streamers and viewers during live broadcasts with adult content from the LiveMe[7] platform, a major Social Live Streaming Service (SLSS). Because the goal was to detect groomer behavior in the conversations using a chat-level representation, the authors considered a subset of chats from the PJ website and investigated if any of the patterns present in the PJ subset was also present in the streaming chats. They started by obtaining the most employed sexual terms in the PJ chats and searched for them in the LiveMe chats using Facebook's FastText library[8] to find similar terms. The authors identified a wide variety of terms with sexual and offensive connotations. They also found out that the use of behavioral features such as emojis and clothing terms is very common to express sexual behavior. As a final related work in the forensic scenario, a two-stage classification strategy using both chat and user level representations was proposed in the work of Fauzi and Bours (2020). The authors presented an ensemble method that trains multiple classifiers and combines their results to evaluate the data. The goal was to first identify the suspicious conversations and then distinguish the victim from the groomer. To make it possible, they prefiltered the data by considering only conversations with exactly two users and with at least 6 messages per user. Then, two ensemble strategies were applied: hard voting, in which each classifier had one vote and the class for a document was the majority voted class, and soft voting, where each classifier computed the probability of each class and the final class for the document was the one with the highest average probability. In the experiments, the authors studied the SGD corpus and found out that their results also surpassed the winner's results Villatoro-Tello *et al.* (2012) at the PAN 2012 competition.

4.1.1 Summary and discussion

In the above section, we have revised related works that are focused on a forensic scenario. A wide variety of tools attempting to confront groomer behavior in online conversations was discussed. Additionally, we identified that most of the proposals employed BFs successfully to learn the communication patterns that a groomer uses. This allows us to confirm that BFs are truly important to the identification of online grooming. Forensic scenario tools presented

[7] <https://www.liveme.com/>

[8] <https://fasttext.cc>

great progress through the past years, however, two main disadvantages lie in the existing approaches. First, there is a lack of analysis in the proposals of the BFs. While it is true that most methods construct BFs aimed at the identification of groomer behavior, at the same time these works simply limit themselves to the use of the features and do not perform any analysis that presents intuitive explanations on why BFs are helpful. For example, many works analyzed the use of sexual connotation terms for the detection of groomer behavior, however, none of these works demonstrated if indeed groomer conversations contain more sexual terms than other conversations. Or for instance, part of the attributes of the SGD corpus is the time at which each message is sent, but none of the proposals was concerned with the analysis of what is the preferred time for groomers to start a conversation. In our humble opinion, the fact that existing works propose BFs without performing any proper analysis is a drawback that limits their correct use. The second major disadvantage is more obvious: none of the tools discussed in this section can be used to prevent online grooming attacks. Unfortunately, these tools can only be applied when the attack has already been performed. Since the detection of groomers should ideally be done in a preventive way, i.e., as soon as possible, and surely before the actual physical contact, it is not possible to wait for all the complete conversations to arrive to trigger an alert. With that in mind, the related works focused on a preventive scenario are described in the following section.

4.2 Preventive scenario

As it was described in Section 4.1, there is a wide variety of forensic tools focused on tackling online grooming, but, unfortunately, these tools are **not useful to prevent an attack**. Therefore, the necessity to develop new tools capable of detecting online grooming in a preventive scenario arises. In this sense, an approach that is gaining increasing interest is the one referred to as Early Text Classification (ETC), which is defined by Burdisso, Errecalde and Montes-y-Gómez (2019) as the task of assigning sequential data to a particular class, as early as possible, and without having a significant loss in terms of accuracy. For López-Monroy *et al.* (2018), ETC consists in effectively identifying the potential risk by using as little text as possible and with as much anticipation as possible. From the aforementioned definitions, one can conclude that the main concern of detecting online grooming in a preventive scenario lies in how quickly a conversation can be noticed as suspicious and how sure the system can be about it.

For this reason, we described in detail the works that are focused on detecting online grooming as quickly and as accurately as possible in a conversation. It is important to note that all the works described in this section make use of the SGD dataset. As it was mentioned in Chapter 3, SGD is the most popular and complete dataset in the context of grooming detection. Another common property observed in works of the preventive scenario is the application of filtering techniques to overcome the data imbalance.

4.2.1 Early Naïve Bayes

Let us begin by outlining the work of Escalante *et al.* (2016), where the classic Naïve Bayes algorithm is adapted to make predictions with little text. They named their proposal Early Naïve Bayes (ENB). The following is a detailed description of this method.

First, the authors represented the dataset with the following equation $D = (d_1, y_1), ...(d_n, y_n)$, where n is the total number of documents and d_j is the j^{th} document with $y_i \in Q$ being its corresponding label, such that $Q = \{Q_1, Q_2, \ldots Q_q\}$ is the set of classes for classification. Then, the objective is to assign the most probable class $\widehat{Q}$ to an unseen document using NB classifier which is briefly described in Section 2.2. We now rewrite the classifier formula with a more similar nomenclature to the one used in the original paper.

$$\widehat{Q} = arg\ max_i\ P(Q_i \mid d_j) \tag{4.1}$$

Here, Equation 4.1 aims to return the most probable class $Q_i \in Q$ given a document d_j. To achieve this, the posterior probability is calculated using Bayes' theorem .

$$P(Q_i \mid d_j) = \frac{P(d_j \mid Q_i)P(Q_i)}{P(d_j)} \tag{4.2}$$

Since the denominator does not affect the decision, it is decided to remove it.

$$P(Q_i \mid d_j) \approx P(d_j \mid Q_i)P(Q_i) \tag{4.3}$$

From here, it is possible to introduce the authors' proposal, the ENB. So we rewrite the above equation (4.3).

$$P(Q_i \mid d_{j,t}) \approx P(d_{j,t} \mid Q_i)P(Q_i) \tag{4.4}$$

Now, in Equation 4.4, a document is depicted by $d_{j,t}$, where t represents the document d_j at time t. Thus, the authors assume that at a time t, they have read t terms of the document. It is even possible to classify an empty document $d_{j,t}$ where $t = 0$, and clearly the probability will be dominated by the priors. Just like that, the authors modified NB to perform early text classification. Then, if we analyze in more detail the components of the NB theorem when using partial information, refer Equation 4.5, we find that now the probability of a document d_j at a time t is decomposed into three components. Where the first component refers to the probability of the class Q_i, the second component is a product $(j : j \in d_{j,t})$ that iterates through the terms j that are present in the document d_j at a time t; and, that the third component is a product $(k : k \notin d_{j,t})$ that is reduced to 1 since the terms k are not present in d_j at a time t. Therefore, for small values of t, the priors dominate the decision, as t increases the content of the document will dominate the other products.

$$P(Q_i \mid d_{j,t}) \approx P(Q_i) \prod_{j:j \in d_{j,t}} P(d_j \mid Q_i) \prod_{k:k \notin d_{j,t}} P(d_k \mid Q_i) \tag{4.5}$$

Finally, to evaluate their ENB proposal with the SGD corpus, the authors represented the documents at a chat level, that is, they represented each conversation as a document, then, they divided the testing set into ten portions containing 10%, 20%, ... and 100% of the information available that is later embedded into a SVM classifier, one by one. This procedure simulates the arrival of messages in an online conversation. Even though the results achieved were not good, to the best of our knowledge, the aforementioned approach is the first work that addresses ETC to detect online grooming conversations.

4.2.2 Profile specific representation

In the work of Escalante *et al.* (2017), a Profile Specific Representation (PSR) for the early recognition of online grooming was proposed. Let us describe this specific related work in detail because one of our contributions in this monograph is to extend/improve the PSR framework. The authors of PSR sought to overcome the bag of words' drawbacks, i.e., high dimensionality and sparsity, by presenting a two-step process. They represented the input dataset at a chat level and described it with the following equation: $D = (d_1, y_1), ...(d_n, y_n)$, where n is the total number of documents and d_i is the i^{th} document with $y_i \in Q$ being its corresponding label, such that $Q = \{Q_1, Q_2, \ldots Q_q\}$ is the set of classes for classification or, more specifically, the profiles of the groomer and the non-groomer classes. The vocabulary is composed of character 3-grams, and it is represented by $V = \{v_1, \ldots v_m\}$, where v_j is a term that occurs in at least one document d_i, and m is the total number of terms in V. In order to represent the terms in a profile space, they are described by vectors that capture the association between terms and target profiles, so each term $v_i \in V$ is described by a vector $t_i = <t_{i,1}, ..., t_{i,q}>$, where $t_{i,j}$ quantifies the association between term v_i and class Q_j. To make it possible, Equation 4.6 is employed.

$$w_{i,k} = \sum_{\forall d_j : y_j = Q_k} log_2 \left(1 + \frac{TF(v_i, d_j)}{len(d_j)}\right) \tag{4.6}$$

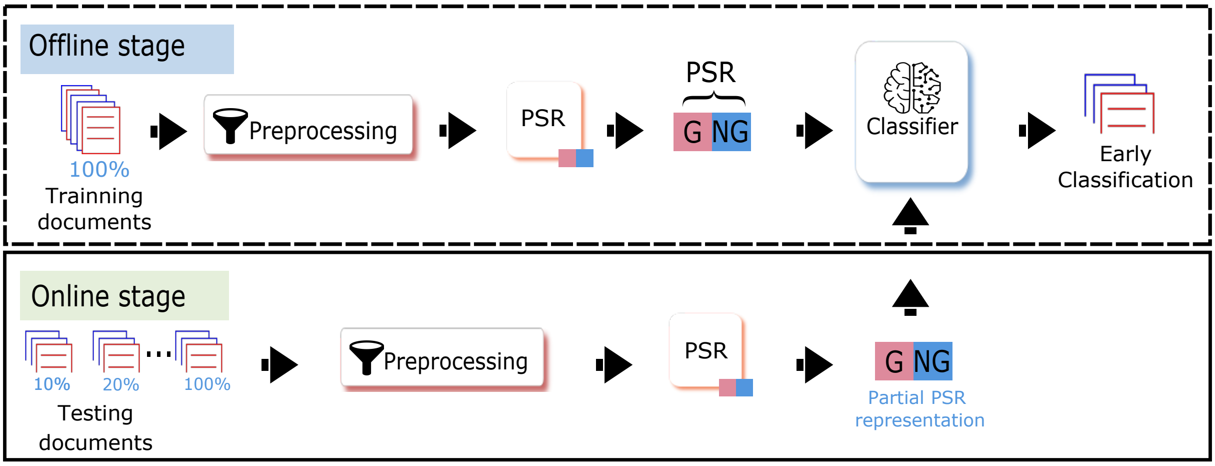
In the equation, for each term $v_i \in V$, a term-frequency-based weighing schema is applied, where $TF(v_i, d_j)$ is the frequency of term v_i in document d_j and $len(d_j)$ is the size of document d_j, while the log_2 function softens the most frequent terms. Then, two normalizations are performed to calculate the final vector t_i.

$$\hat{t}_{i,k} = \frac{w_{i,k}}{\sum_{i=1}^{|V|} w_{i,k}} \tag{4.7}$$

$$t_{i,k} = \frac{\hat{t}_{i,k}}{\sum_{k=1}^{|Q|} w_{i,k}} \tag{4.8}$$

The first one (Equation 4.7) considers the proportion of the $|V| = m$ terms in each class and the second one (Equation 4.8) normalizes the weights computed for the $|Q| = q$ classes to make the weights w_i comparable among the classes. In this way, each unique term v_i is described by a vector that contains its relevance to each class/profile. Therefore, the first part of the two-step

Figure 9 – Pipeline of the PSR framework based on (ESCALANTE *et al.*, 2017). The chart depicts both the offline (dashed line) and the online (solid line) stages. The offline stage preprocesses the training documents and represents them in a PSR vector space; then, the classifier is trained with this representation. In the online stage, partial PSR vector representations are obtained for the partial documents and they are given to the trained classifier to perform early predictions.



process is complete.

$$PSR(d_k) = \sum_{v_i \in d_k} \beta \cdot t_i \tag{4.9}$$

Later, to address the second step, i.e., to represent the documents with vectors, all the terms present in each document d_k are extracted and then a weighted sum of their vectors is computed by following Equation 4.9, where β is a scalar that weights the relevance of terms to the document; in their paper, $\beta = 1$ is used for all experiments. Thus, each document is represented through a non-sparse and low-dimensional vector whose size is the number of classes $|Q| = q$. To evaluate the method in an ETC scenario, the testing set of the **filtered** SGD corpus was divided into portions of text containing $10\%, 20\%, \ldots$ and 100% of each conversation, and, as a result, their score surpassed the one reported in Escalante *et al.* (2016). Figure 9 depicts the pipeline of PSR. In the offline stage, the documents are preprocessed. Then, the PSR vector representation is extracted from the clean documents containing the profiles for the groomer and non-groomer conversations; see the *G* and *NG* squares in the pipeline. The chosen classifier is then trained with this representation. At the testing time, the online stage preprocesses partial documents to extract partial PSR vectors from them. These vectors are embedded into the trained classifier to perform predictions with partial information.

4.2.3 Multi-resolution representation

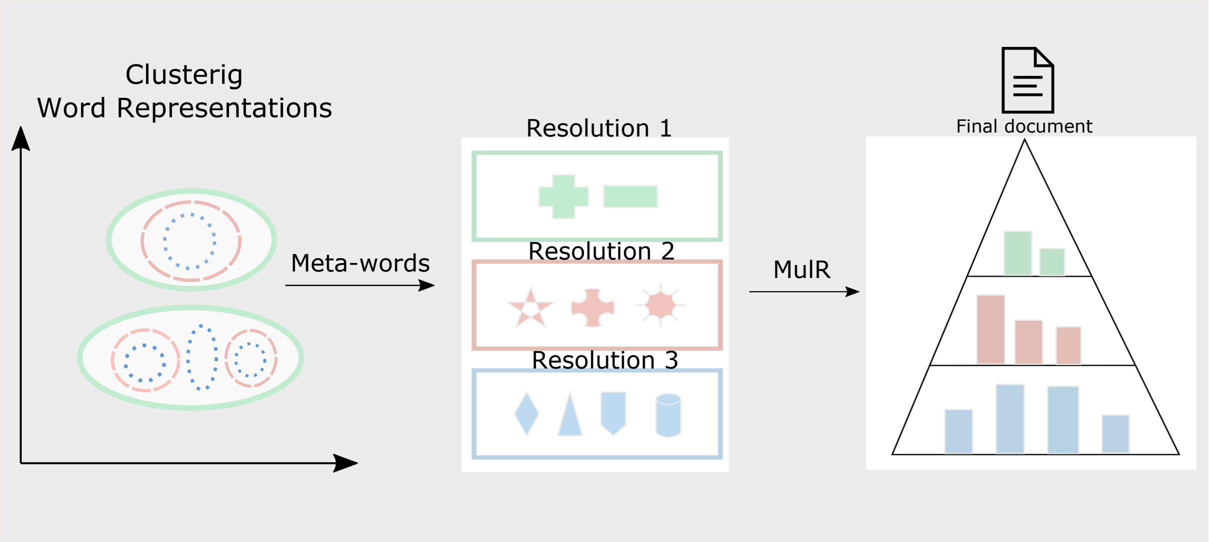
Recently, a novel approach called Multi-resolution Representation (MulR) was presented in López-Monroy *et al.* (2018). It allows to generate multiple "views" of the text so to capture different semantic meanings for words and documents at different levels of granularity. To do that, the authors first introduce the concept of "a single resolution: Bag of Centroids (BoC)", where similar to the ENB and PSR methods, the input dataset was represented at a chat level and described it with the following equation: $D = \{(d_1, y_1), \ldots (d_n, y_n)\}$, where n is the total number of documents and d_i is the i^{th} document with its corresponding label $y_i \in Q$, such that $Q = \{Q_1, Q_2, \ldots Q_q\}$ is the set of classes for classification. Then, it is computed the vector representation v_i of each word w_i in the vocabulary. For this specific framework, the vocabulary is composed of only words, and it is represented by $V = \{w_1, \ldots w_r\}$, where w_j is a word that occurs in at least one document d_i, and r is the total number of unique words in V. The vector representation of each word $v_i \in V$ is done through well-known word embeddings such as Word to Vector (W2V) (MIKOLOV *et al.*, 2013), PSR (ESCALANTE *et al.*, 2017) and Temporal Variations of Terms (TVT) (ERRECALDE *et al.*, 2017). Following, each word embedding v_i is grouped by distance, and the cluster centers are found to create the proposed meta-words. The algorithm k-means is employed for the clustering process. The found centers are described with the following equation: $C = \{c_1, c_2, \ldots, c_k\}$, with k being the number of selected centroids. Thus, each v_i can be represented by one of the centroids $c_i \in C$. The choice centroid is the one that has the minimum distance with v_i . The same logic is employed with the BoC algorithm, where a whole document is represented through centroids/meta-words. The authors also note that the number of centroids k serves to control the coarseness of the dataset. Thus, if each word becomes a centroid, the resulting representation is equivalent to the typical BoW representation, whereas a coarser representation, with only one meta-word, will be equivalent to having the average meta-word of the entire collection. Later, the variation of this method called MulR is introduced. The algorithm takes advantage of the coarse property; and, in a simplified form, the method aims to take advantage of the different levels of granularity. The MulR algorithm can be mathematically depicted by Equation 4.10, where d_j correspond to the multi-resolution representation of the j^{th} document with $\{k_1, k_2, \ldots k_n\}$ granular levels. Let's remember that $BoC_{k_i}(d_j)$ represents the j^{th} document through a single resolution with $k = i$ centroids.

$$MulR(d_j) = BoC_{k_1}(d_j) \cup BoC_{k_2}(d_j) \cup \ldots \cup BoC_{k_n}(d_j) \tag{4.10}$$

Through the results, it is confirmed that MulR improves the performance of a single resolution BoC by combining information at various granularity levels. This whole process can be observed in Figure 10. Where the different BoC resolutions are characterized by different meta-words and the combination of each resolution gives as result the overall Multi-resolution representation.

MulR was later tested with the SGD testing dataset divided into ten portions of text available. After the evaluation, the method's results surpassed the ones achieved by both ENB (ESCALANTE *et al.*, 2016) and PSR (ESCALANTE *et al.*, 2017), and their scores represented the state-of-the-art for the **filtered** SGD corpus. In our experiments, we take the PSR and the three variants of method MulR, i.e., MulR with three different words embeddings (W2V, PSR, TVT), as concurrent techniques in the area of early detection of grooming to compare it with our

Figure 10 – Pipeline of the MulR framework. First, the algorithm represents the documents as meta-words using three hypothetical resolutions. Then, the MulR algorithm is the combination of each resolution. Based on (LÓPEZ-MONROY *et al.*, 2018).



proposal; then, we demonstrate that our results are currently the state-of-the-art for the **complete** SGD corpus.

4.2.4 Summary and discussion

In the above section, we have described the most relevant works that belong to a preventive scenario. They all attempt to detect groomer behavior in online conversations as early and as accurately as possible. However, some disadvantages are evident in the existing proposals, starting with the fact that compared with the forensic scenario, works under this approach are scarce and the necessity to develop more preventive tools is not being fulfilled. It is also evident that all the existent works use filtering techniques to overcome the data imbalance, and, as a consequence, they artificially soften the difficulties that are inherent to the real-world problem. Because we are dealing with a system focused on preventing a crucial risk, filtering out any suspicious conversation means not analyzing chats with groomer content that could hurt a child, and this is not acceptable in real-life applications. Instead, all conversations must be analyzed when looking for groomer patterns. For example, chats with only one participant that are commonly removed by the filtering techniques may contain offensive text written by a groomer, reason enough for which the conversation does not receive a response and it is precisely this type of conversation that should generate an alert by the system. Another consequence of the use of filtering techniques is the accuracy of results reported in the literature, which does not take into account the errors referring to the groomer conversations that were filtered out and **never analyzed** by the classifier, so these results may be considered artificial since the accuracy reported could not be actually obtained in a real system. We argue that filtered-out groomer conversations should be reported as errors of the system; otherwise, in our humble opinion, the results are being masked and do not represent the real accuracy achieved by the proposed methods. Besides, the possibility that the methods do not perform well with a large amount of unbalanced data is another relevant drawback. In our proposal, no filtering process is applied and some fundamental adjustments to the original PSR vector representation are carried out, like the preprocessing type and the weighting schema, in such a way that they make our method suitable for a large amount of disproportionate data.

Another notable limitation is the absence of behavioral features in methods for the preventive scenario. To the best of our knowledge, none of the existing works employ behavioral features to detect grooming behavior in conversations, which is a gap in the literature since the importance of these features was already confirmed in the forensic scenario. To tackle this issue, part of our contributions in this monograph is to conduct a detailed analysis to understand the background of the seven behavioral features that we employ in our framework, i.e., number of participants in a conversation, emoticons, correctly-spelled words, sexual topic words, the time when a conversation starts, intervention words per user, and sentiment and emotional markers. Four of these features, i.e., emoticons, correctly-spelled words, sexual topic words, and sentiment and emotional markers, have already been used to detect grooming in the forensic scenario, while, to the best of our knowledge, the others have never been used in groomer detection. Nevertheless, they have been helpful in different real-life data-mining applications, such as: a) the number of participants in a recorded conversation was used to classify Alzheimer's dementia speech (la Fuente Garcia; Haider; Luz, 2020), and we adapt this feature to identify if the groomer interacts with multiple participants in a conversation or just with the victim; b) time was used as a feature to evaluate the risk of flooding in urban environments (BERNARDINI *et al.*, 2017), and we adapt this feature to understand the schedules that groomers employ to approach their victims, and; c) the average number of words per message was used to identify malicious emails (MARTIN *et al.*, 2005), and we adapt this feature to calculate the average number of words that different participants write when they intervene in a conversation. Therefore, in our proposal we aim to:

i) reinforce, for the preventive scenario, the benefits of some behavioral features that were already used in the forensic scenario, and;

ii) propose to take advantage of features that, to the best of our knowledge, have never been used before to spot groomer behavior.

Table 14 summarizes the related works for both the forensic and the preventive scenarios. The columns describe: the type of document representation used, i.e., user level or chat level; whether or not any filtering technique was applied; if behavioral features were used; the datasets employed, and; the scenario considered. From the table, important characteristics can be observed. For instance, the most frequently used dataset is the SGD corpus and its use is more popular among the preventive works. In fact, to the best of our knowledge, the SGD corpus is the only dataset

Work	Document representation	Filtering process	Behavioral features	Datasets	Scenario
Pendar et al., 2007 (PENDAR, 2007)	user level	no	no	PJ chats	forensic
Edwards et al., 2009 (EDWARDS; LEATHERMAN, 2009)	chat level	no	yes	PJ chats	forensic
McGhee et al., 2011 (MCGHEE *et al.*, 2011)	chat level	no	yes	PJ chats	forensic
Kontostathis et al., 2012 (KONTOSTATHIS *et al.*, 2012)	user level	yes	yes	SGD	forensic
Morris et al., 2012 (MORRIS; HIRST, 2012)	user level	no	yes	SGD	forensic
Parapar et al., 2012 (PARAPAR; LOSADA; BARREIRO, 2012)	user level	no	yes	SGD	forensic
Villatoro-Tello et al., 2012 (VILLATORO-TELLO *et al.*, 2012)	chat and user level	yes	no	SGD	forensic
Cheong et al., 2015 (Cheong *et al.*, 2015)	chat level	no	yes	Movie star planet chats	forensic
Bogdanova et al., 2016 (BOGDANOVA; ROSSO; SOLORIO, 2014)	chat level	no	yes	PJ + cybersex + NPS chats	forensic
Ebrahimi et al., 2016(EBRAHIMI; SUEN; ORMANDJIEVA, 2016)	chat level	no	no	SGD	forensic
Cardei et al., 2017 (CARDEI; REBEDEA, 2017)	chat and user level	yes	yes	SGD	forensic
Zuo et al., 2018 (ZUO *et al.*, 2018)	chat and user level	no	no	Pan 2013 subset	forensic
Lykousas et al., 2016 (LYKOUSAS; PATSAKIS, 2020)	chat and user level	no	yes	Live me chats	forensic
Fauzi et al., 2020 (Fauzi; Bours, 2020)	chat and user level	yes	no	SGD	forensic
Escalante et al., 2016 (ESCALANTE *et al.*, 2016)	chat level	yes	no	SGD	preventive
Escalante et al., 2017 (ESCALANTE *et al.*, 2017)	chat level	yes	no	SGD	preventive
López-Monroy et al., 2018 (LÓPEZ-MONROY *et al.*, 2018)	chat level	yes	no	SGD	preventive
Our proposal	chat level	**no**	**yes**	SGD; PJZ; PJZC	preventive

Table 14 – Summary of approaches for the forensic and the preventive scenarios. We report for each method: the type of document representation; whether or not a filtering technique was used; if behavioral features were employed; the datasets used, and; the scenario considered.

ever studied for the preventive scenario. On the other hand, in forensic works, other datasets are beginning to appear, which in the future may help to address the lack of datasets in the literature of grooming detection. The table also confirms the frequent use of filtering techniques and the absence of behavioral features in the preventive works. For these reasons, with our proposal, we want to establish a notable difference from the other preventive works, since, to the best of our knowledge, we are the **first ones** to employ behavioral features in the area of early detection of grooming. Besides, because we want our proposal to be promptly applicable to real-life cases, we do not use any filtering technique when preprocessing the input data. Therefore, to the best of our knowledge, we are also the **first ones** to perform early classification of grooming in unfiltered data by reporting accuracy-related results that consider every single conversation. Finally, it can be seen from the table that we do not limit ourselves to only using one dataset for the evaluation of our proposal. Instead, we employ two new datasets, i.e., the PJZ and the PJZC datasets. So, we end up offering one more contribution by being the **first ones**, in the preventive scenario, to use a dataset different from the SGD corpus.

We believe it is important to mention that we tried to use the existing datasets in the forensic scenario and apply them in the preventive scenario. However, we encountered several difficulties at the time of their use, for example, some had policies of a non-disclosure agreement, others did not have their labels available, and with a few, we are still waiting for a response. Given all these difficulties in using the forensic datasets, we decided to assemble our own datasets which we name as PJZ and PJZC. The PJZ dataset is composed of groomer conversations extracted from the PJ website[9] and non-groomer conversations extracted from IRC logs through the #ZIG channel[10]. Both sets of conversations are completely disjoint from those conversations in the SGD dataset. To be specific, the collected PJ conversations are from the year 2013 onwards, chats occurring one year after the publication of the SGD corpus. And the #ZIG set is collected

[9] <www.perverted-justice.com/?con=full>
[10] <https://github.com/marler8997/zig-irc-logs>

from the year 2017 onwards using the IRC channel. Then, for the composition of the PJZC dataset, we decided to add, to the PJZ dataset, conversations with more general topics. To do that, we include chats from the Chit-Chat dataset (MYERS; ETCHART; FULDA, 2020), which contain non-offensive messages exchanges between university students. In further sections, we will describe these assemblies in more detail.

The following chapter outlines our first main contribution. We present the seven behavioral features proposed in this work. And, we also describe in detail one study conducted to understand the reasons why each feature contributes to the task of early detection of grooming.

CHAPTER 5

PROPOSED BEHAVIORAL FEATURES

This chapter presents our first major contribution: a set of seven Behavioral Features (BFs) to be used for the early detection of grooming in online text conversations. Three of these features were never used before to reveal groomers, while the others have been used only in the forensic scenario, but no intuitive explanation for the reasons why they are helpful to this task is presented in the literature. Here, besides describing the features themselves, we justify their use by presenting a detailed study aimed at understanding the reasons why each of the seven features contributes to detect groomer behavior. In this way, we tackle the lack of analysis regarding the BFs that were already employed in the forensic scenario, and, at the same time, we shorten the gap existing between the forensic works and the works of the preventive scenario, since, to the best of our knowledge, we are the first ones to employ BFs for the early detection of grooming. Furthermore, please note that all analyses performed for the identification of the proposed BFs were done using only the **training data** from the SGD dataset. The following subsections describe and analyze in detail each behavioral feature that we propose to use in our framework.

5.1 Number of participants in a conversation

The number of participants is a behavioral feature that has never been used before for grooming detection. This feature is not explicitly in the attributes of the SGD dataset, yet, to compute its value for a conversation is simple, since one only needs to count the number of unique participants. Let us use the training instances of the SGD corpus to illustrate why this feature helps to distinguish between groomer and non-groomer conversations. Figure 11 reports the percentages of conversations from classes *G* and *NG* according to the number of participants. We consider three distinct categories of conversations:

i) Group: conversations where more than two participants interact;

Figure 11 – Percentage of groomer and non-groomer chats in the SGD training set according to the number of participants. The x-axis distinguishes group conversations, monologues, and pair conversations, while the y-axis reports the corresponding percentages of groomer and non-groomer chats. Note that the occurrences of monologues and pair conversations are nearly balanced for class *G*, and not for class *NG*.

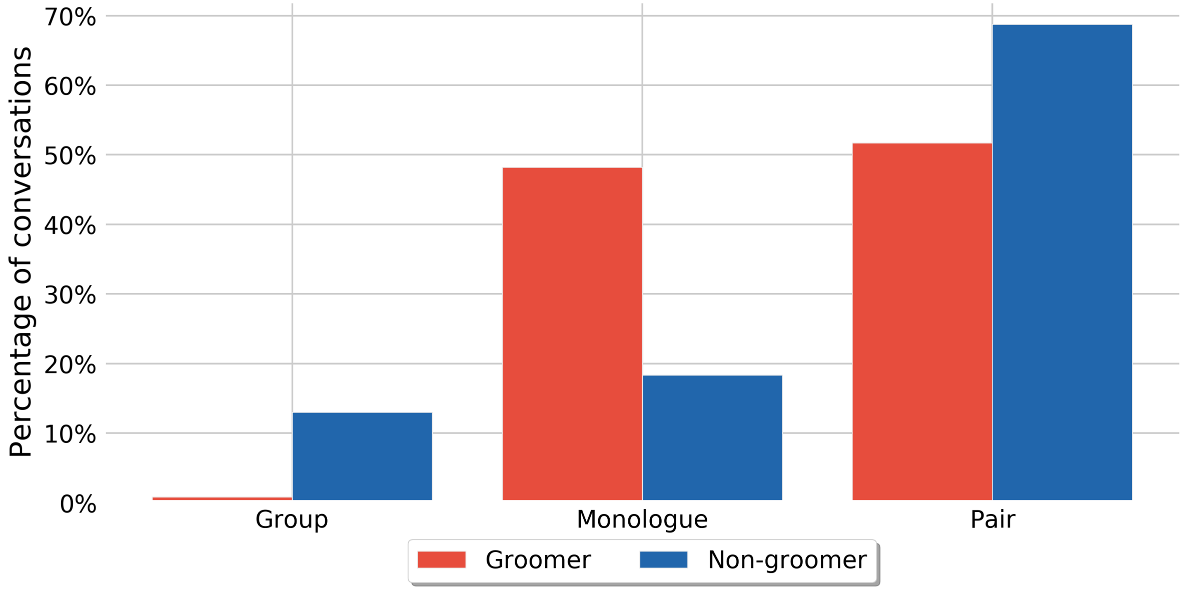
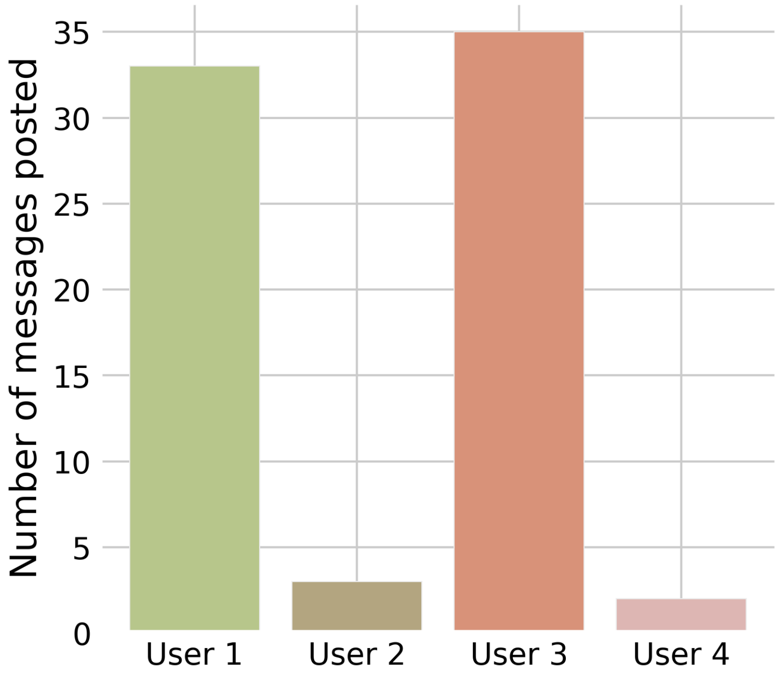


Figure 12 – Number of messages posted per user in a group conversation of class *G*. Note that only Users 1 and 3 interact constantly, so it is similar to a pair conversation, which is the category of conversation preferred by groomers.



ii) Monologue: conversations with a single participant;

iii) Pair: conversations between exactly two participants.

Overall, it is easy to note that there are larges percentages of pair conversations in both classes, which is not surprising since conversations between two people are indeed very common. However, the percentages of pair conversations reveal an interesting pattern: pair conversations are more common in class *NG* than they are in class *G*, this gives us information that may help us with the class distinction. Interestingly, the opposite pattern occurs when we focus our attention on the monologues: they are considerably more common in the groomer conversations than in the non-groomer ones. In fact, monologues are almost as common as pair conversations for class *G*, despite the fact that they are usually filtered out of the analysis by the previous works. These percentage values make us believe that the groomer monologues are failed attempts by the groomers to contact their victims, for example, in cases where a groomer explicitly wrote an offensive message and the victim did not respond. Let us use Table 15 to exemplify this situation by reporting the contents of seven groomer monologues from the SGD training data. In our opinion, Conversations 4 and 6 are inconceivable to happen between an adult and a child, as the groomer tries to convince the victim to meet offline in the former case, while, in the latter case, the groomer displays sexual behavior and receives no response. These are examples of conversations that state-of-the-art methods, such as PSR and MulR, filter out of the analysis, when, in fact, they should generate high-level alerts in the system.

Table 15 – Examples of groomer monologues from the SGD training data.

Conversation Id	Author Id	Time	Content
1	A	02:18 AM	you there? hello?
2	A	03:24 AM	hey luv, sorry i wasnt on just got back i try b on 2day at night, miss ya and luv u. Ty
3	A	01:16 AM	I miss ya
4	B	11:18 PM	know what I thought of you could get your mom to take you to see a movie matinee today and tell her you were going to meet a girl friend and i could meet you. If you want you can go ahead and do it and call me on the payphone when you get there and I will look for your call today...just an idea let me know.
5	C	04:11 PM	whats up
6	D	07:11 PM	hey sexy how are u doing was wonder if u mind telling me your favorit fantasy
7	E	04:29 PM	:-*

Additionally, it can be noted that group conversations are the ones that occur more rarely, especially in class *G* where they practically do not exist. From this characteristic, it is clear that the groomers prefer to isolate their victims to achieve their goal. Let us report evidence to support this hypothesis by analyzing in deep the group conversations from the SGD training set. There are more than $8,000$ group conversations in class NG, where sometimes more than 30 participants avidly interact in the same chat. On the other hand, only 5 group conversations exist in class G, and they all look like pair conversations since there are always two participants interacting with great enthusiasm, while the others rarely post a message. Figure 12 reports the number of messages posted per user in one of these conversations. As it can be seen, there are four participants in total, but only two of them interact constantly; the rest hardly contribute to the chat. The same pattern occurs in the other 4 group conversations of class G, thus corroborating our understanding that groomers prefer to isolate their victims in pair conversations.

Therefore, we propose to employ the number of participants in a conversation as a behavioral feature. To the best of our knowledge, it has never been used before to reveal groomers. According to the aforementioned analysis, the number of participants provides great value in highlighting the different patterns between groomer and non-groomer conversations. In general, it is expected for class G to have nearly no group conversations and a balanced quantity of monologues and pair conversations. On the other hand, we expect for class NG to have much more pair conversations than monologues and group conversations, with nearly balanced quantities of these last two categories of conversation.

5.2 Sentiment and emotional markers

Sentiment and emotional markers are also considered as behavioral features in our framework. Let us remember that BFs attempt to capture how an author uses a word instead of focusing on what are the words used. Thus, the features described here aim to identify the emotions and sentiments that are the most meaningful to groomer conversations and distinguish them from what is meaningful to non-groomer ones. For us, sentiment and emotional markers are closely related to the discourse model that groomers employ to nurture intimacy with their victims, which was described in Chapter 4. In this case, some research questions could be postulated regarding the sentiment markers, e.g., “do groomer conversations contain more negative words than non-groomer ones?”. Or, should it be on the contrary because the groomers’ objective is to get close to their victims: “do groomer conversations contain more positive words than non-groomer ones?”. Similar questions can also be postulated for the emotional markers, e.g., “do fear or joy predominates in groomer conversations?” and “is the predominant emotions distinct in non-groomer chats?”. With that in mind, we propose to employ two emotional lexicons that were never used before for grooming detection:

i) NRC emotion lexicon[1]: a lexicon associated with eight basic emotions, i.e., anger, fear, anticipation, trust, surprise, sadness, joy, and disgust, besides two sentiments, i.e., negative

[1] <https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm>

Figure 13 – Percentage of words in conversations from the SGD training set that are related to a specific emotion or sentiment defined in the NRC emotion lexicon. Class *G* is represented by left-sided bars with a solid border; right-sided bars with a dotted border represent class *NG*. The emotions highlighted are the ones that best distinguish the classes, i.e., anger (🟥), anticipation (🟪), joy(🟩), positive (🟩), sadness (🟧) and surprise (🟨). They are used as behavioral features in our framework.

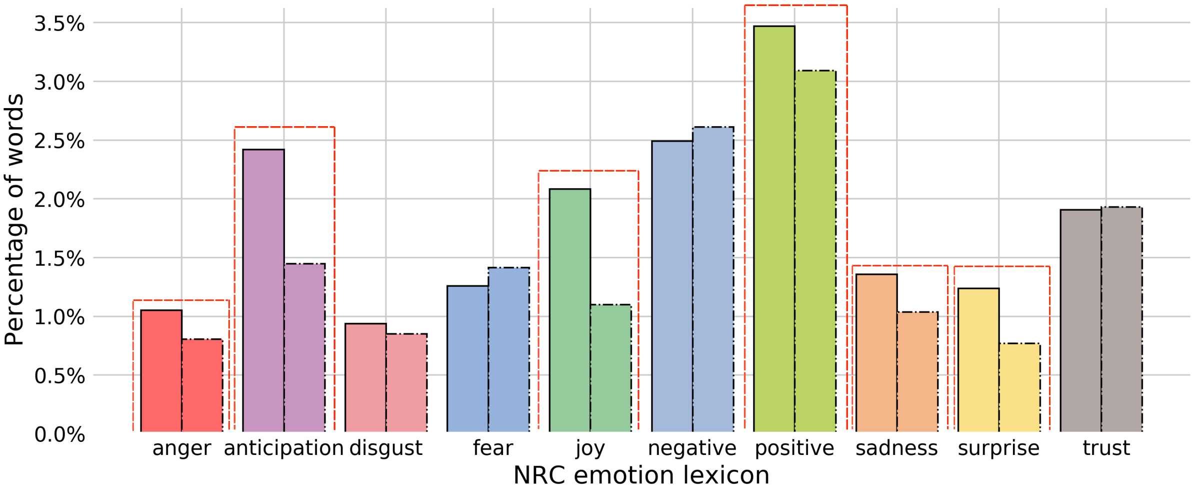
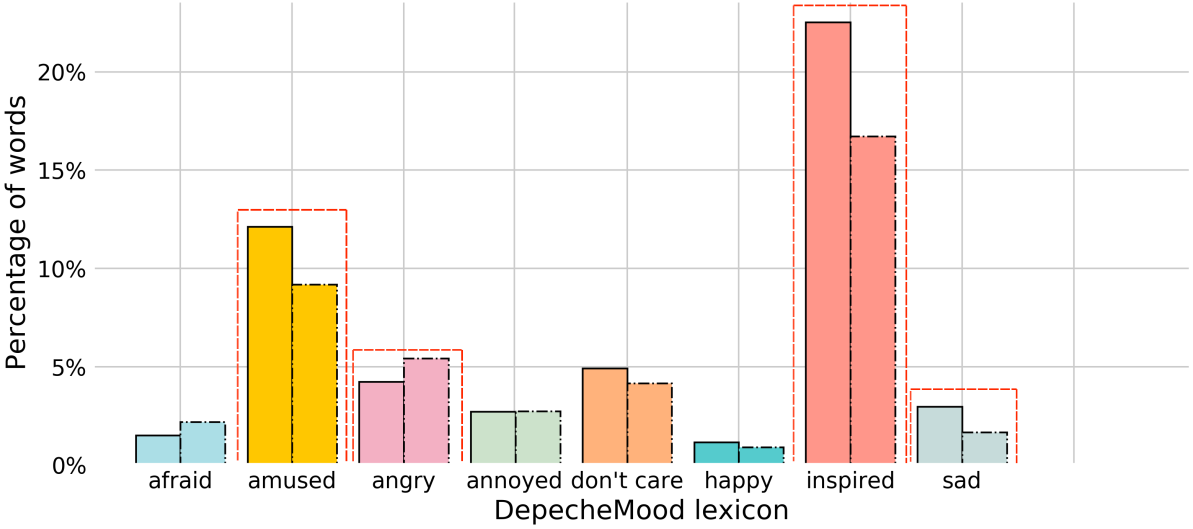


Figure 14 – Percentage of words in conversations from the SGD training set that are related to a specific emotion defined in the DepecheMood lexicon. Class *G* is represented by left-sided bars with a solid border; right-sided bars with a dotted border represent class *NG*. The emotions highlighted are the ones that best distinguish the classes, i.e., amused (🟨), angry (🟪), inspired (🟥) and sad (⬜). They are used as behavioral features in our framework.



and positive;

ii) DepecheMood lexicon[2]: a lexicon associated with eight emotions, i.e., afraid, amused, angry, annoyed, don't care, happy, inspired, and sad.

[2] <https://www.aclweb.org/anthology/P14-2070/>

To make it possible, we studied the SGD training data aimed at identifying the emotions and sentiments that best distinguish groomer and non-groomer behavior. Therefore, a word-counting process was performed considering each emotion and sentiment of the lexicons individually. The results obtained are reported in Figures 13 and 14; they refer respectively to lexicons NRC emotion and DepecheMood. We report the percentage of words that are linked to one emotion or sentiment with regard to the total number of words present in conversations of each of the classes. As it can be observed, there are emotions or sentiments in both lexicons that produce considerable contrast; this information may help us with the distinction of *G* and *NG* classes.

According to the NRC lexicon, groomers employ positive words more frequently than non-groomers; see the light-green (🟩) bars in Figure 13. We believe that it occurs mostly in chats that received responses from the victims, in cases when a groomer started a conversation with positive words to establish a trusting relationship. We also found out that words related to joy (🟩) and anticipation (🟪) occur more frequently in the groomer conversations than in the non-groomer ones; provided that these emotions may be understood as positive sentiments, we believe that they were also evoked by the groomers to establish trust at the beginning of the conversations. Emotions such as anger (🟥), sadness (🟧) and surprise (🟨) are also considerably more common in class *G* than in class *NG*, despite the fact that they are used with relatively little frequency in both classes. We believe that these emotions may be mainly embedded in groomer monologues with offensive connotations, as these chats tend to have little text, which would explain the low frequencies. Finally, for disgust (🟥), fear (🟦), negative (🟦) and trust (⬜), we did not find much distinction between the classes.

When considering the DepecheMood lexicon, emotions amused (🟨) and inspired (🟥) are considerably more common in the groomer conversations than in the non-groomer ones. These are two positive emotions that may reflect the deceptive trust development phase of the groomer discourse model. Additionally, despite not being commonly employed, emotions angry (🟥) and sad (⬜) also have contrasting usages between the classes. Emotion angry (🟥) occurs more frequently in non-groomer conversations than in the groomer ones, while the opposite pattern is observed for sad (⬜). Finally, the remaining emotions do not contribute much to differentiate the classes; they are: afraid (🟦), annoyed (🟩), don't care (🟧) and happy (🟦).

From the aforementioned analysis, there are clear evidences that groomer conversations tend to be more "full of emotions" than non-groomer conversations. We believe that this "avalanche" of emotional words is commonly used by groomers both to create a trusting relationship with the victim and also to convince the child to meet offline and begin a sexual relationship. With that in mind, we propose to take advantage of emotional and sentiment markers by using them as behavioral features in our framework. Specifically, a collection of ten behavioral features is used with this purpose, in which we describe the proportions of use of the most discriminative sentiments and emotions of each lexicon. From the NRC emotion lexicon, we use anger (🟥), anticipation (🟪), joy (🟩), positive (🟩), sadness (🟧) and surprise (🟨) as meaningful features. And,

Figure 15 – Cloud of the most frequent words related to the sexual topic that occur in the SGD training set.

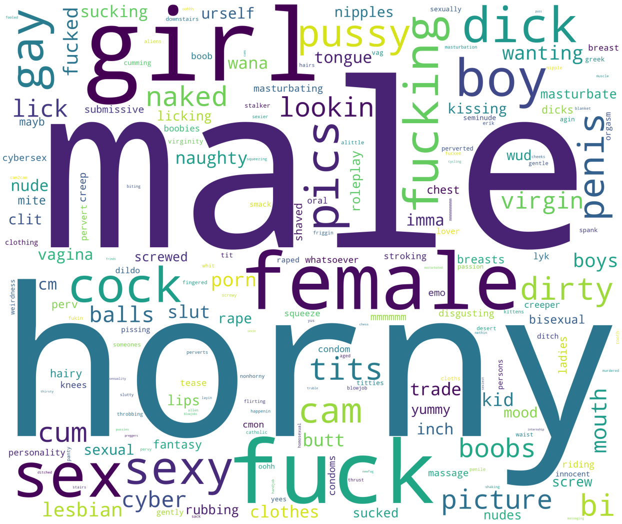
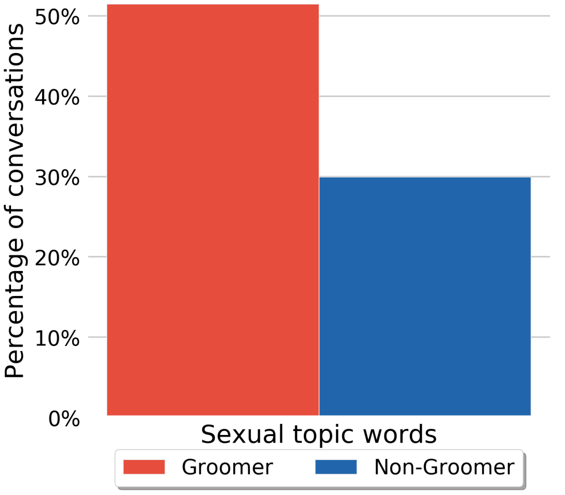


Figure 16 – Proportions of G and NG conversations that are associated with sex-related words in the SGD training set.



from the DepecheMood lexicon, we use amused (🟨), angry (🟪), inspired (🟧) and sad (⬜).

5.3 Sexual topic words

Words with sexual connotation are commonly used to create BFs in tools of the forensic scenario, so they tend to be relevant also to the preventive scenario. As it was described in Chapter 4, the discourse model indicates that a phase of sexual gratification commonly occurs once the groomer establishes a trusting relationship with the child. In this sense, we believe that one way to capture such behavior is by identifying the use of words with sexual connotation that should never happen between a child and an adult. To create a behavioral feature that describes this pattern, we manually selected a small collection of correctly-spelled words that are related to the sexual topic. Let us name this collection as `sex_words`. Then, because it is likely that many sexual words written by actual groomers will be intentionally misspelled, collection `sex_words` was extended by using Facebook's FastText library[3] to create subword-informed word representations from the SGD training data. As a result, the closest neighbors of each correctly-spelled word were identified and added to collection `sex_words` because they are probably misspelled versions of that word. As an example, some of the closest neighbors of the correctly-spelled word "dick" are "dickk", "dickkk", and "dickss"; they were then added to the final collection of words with sexual connotation.

Figure 15 illustrates the sex-related words that are most frequently used in the conversations. It reports a cloud of words created by counting how many times each word of `sex_words` occurs in the SGD training data. As it can be observed, there are numerous misspelled and abbreviated words. Besides, there are words in the collection that were incorrectly identified with the FastText

[3] <https://fasttext.cc>

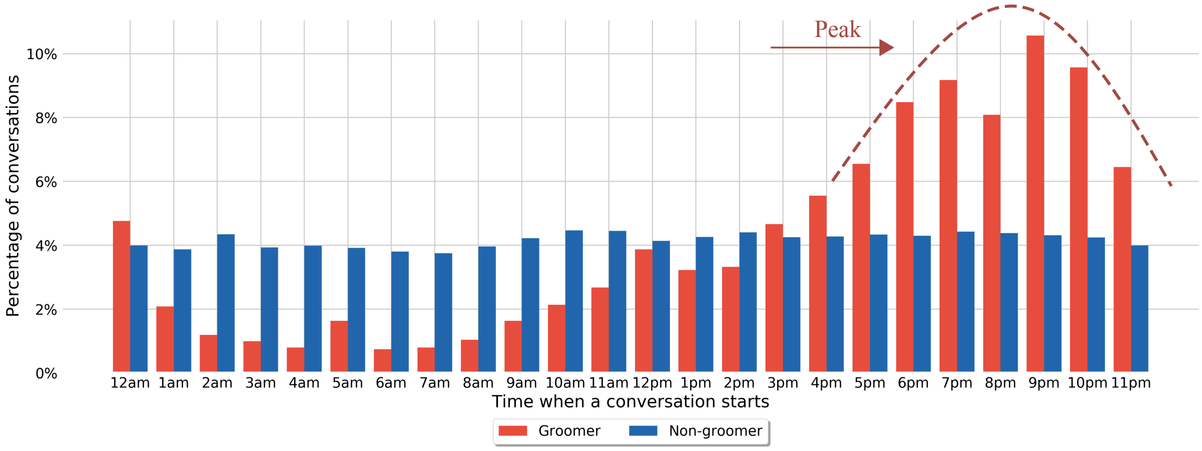
library, such as “erick”, “persons” and “aliens”. These words were manually removed from `sex_words`. After this final adjustment, we decided to investigate if the sexual topic words were employed more frequently in the groomer conversations than in the non-groomer ones. Figure 16 reports the proportions of groomer and non-groomer conversations from the SGD training set that includes any sex-related word. It turns out that nearly half of the groomer conversations include words of the sexual topic, while only $\sim 30\%$ of the non-groomer conversations have such type of words. Therefore, there is clear evidence that the presence of sex-related words indeed highlights the differences between the classes, and we propose to take advantage of this fact by using the proportion of sex-related words as one of the behavioral features that describe a conversation in our framework.

5.4 Time when a conversation starts

The time when a conversation starts is a behavioral feature that has never been used before to detect grooming. Although it is an integral part of the SGD corpus, to the best of our knowledge, this feature has been completely ignored so far in the literature. With that in mind, we have decided to investigate whether or not there is a preferred schedule for groomers to start a conversation with their victims, and, if so, to take advantage of it aimed at preventing this risk. As it happened with the analysis of the sentiment and emotional markers, some research questions could be postulated on this matter. For example, “do groomers prefer to initiate a conversation at night?”, or “do they have other preferences?”, “what about the preferences of non-groomer users?”, and “are there specific moments when victims most access computers, tablets or smartphones, thus being more exposed?”. To answer such questions, we have analyzed the hour of the first message posted in each conversation from the SGD training set.

Figure 17 has the results of this analysis. It reports the percentage of groomer and non-groomer conversations from the SGD training set according to the hour of the first message posted. Red and blue bars represent groomer and non-groomer conversations, respectively. Note that the groomer conversations begin mostly at night, with a clear peak around 9pm that indicates a risky workflow of groomers and children interactions. We believe that these results fit very well the victims’ school schedule. Let us remember that while it is true that the victims in the SGD dataset are volunteers pretending to be children, it is also true that the groomers were unaware of this fact, so one may expect that they tended to approach their victims at the moments when children had more access to computers, which is probably after the school period. It can also be observed that in the early morning hours, such as from 1am to 6am, there are very few groomer conversations. This is probably because most children go to rest during this period. On the other hand, there is a quasi-uniform distribution in the starting hour of the non-groomer conversations. It suggests that these are conversations that occurred mostly between adults with different work schedules and with more access to devices that allow digital communication. To conclude, let us highlight that the time when a conversation starts has never been used before to reveal groomers. Despite this

Figure 17 – Percentage of groomer and non-groomer chats from the SGD training set according to the hour of the first message posted. Red and blue bars represent groomer and non-groomer chats, respectively. Note that there is a quasi-uniform distribution in the starting hour of non-groomer chats; it contrasts with the groomer chats that begin mostly at night, with a clear peak around 9pm that indicates a risky workflow of groomers and children interactions.



fact, the aforementioned analysis provides a clear indication that it offers considerable contrast between groomer and non-groomer chats. Therefore, we propose to take advantage of this fact by using the hour and the minute of the first message posted as two of the behavioral features that describe a conversation in our framework.

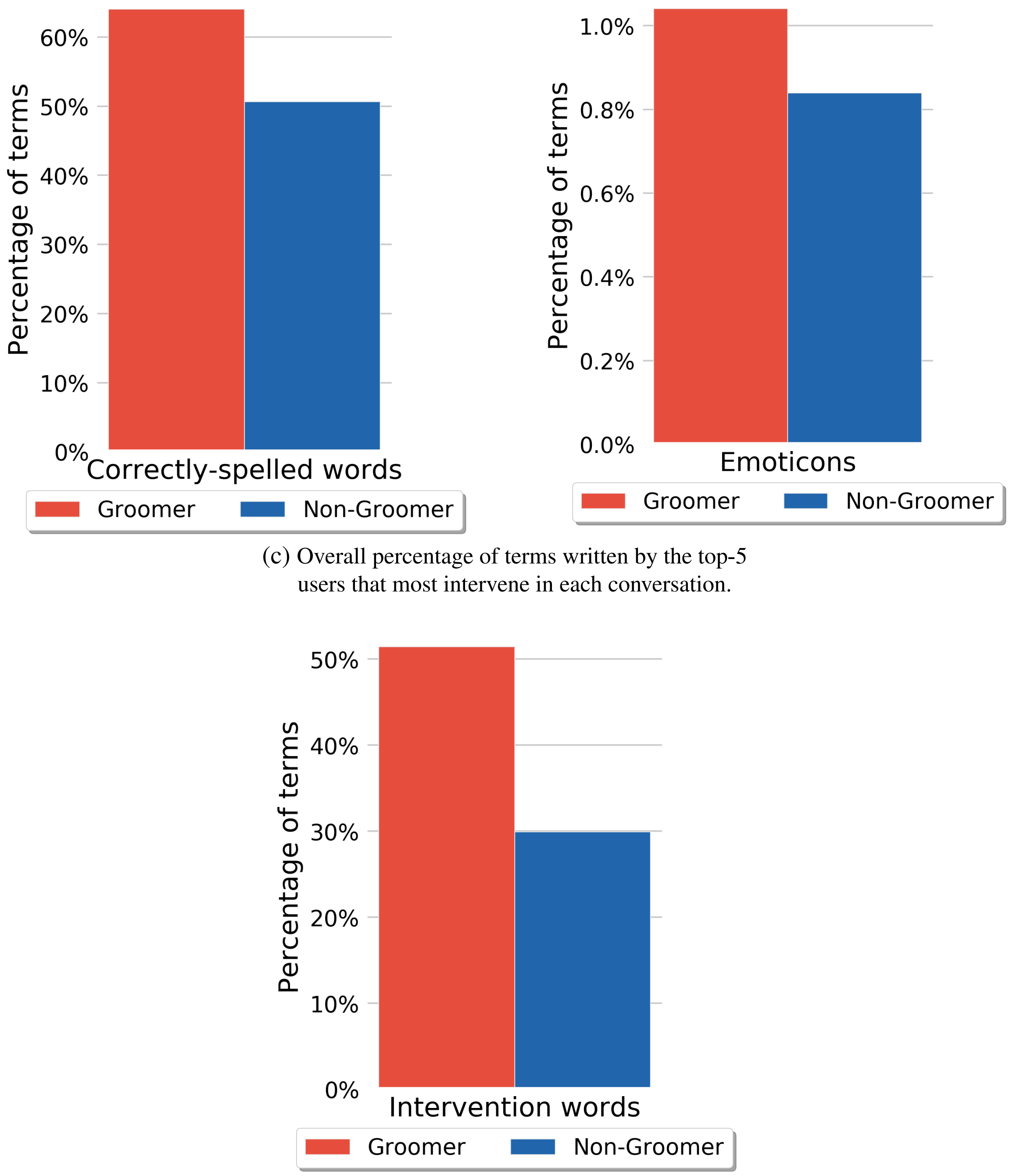
5.5 Writing style

The writing style of a user may help to reveal groomer behavior. In general, the style can be described by different elements, some of which may be more useful than others for our target purpose. For example, unnecessary capital letters, e.g., “CALL ME”, words with excessive length, e.g., “hiiiiii”, and redundant punctuation, e.g., “help!!!!”, are usually employed to accentuate emotions, but the analysis of their occurrences may not help much since they are common in both classes. With that in mind, we propose to use three elements of writing style to build behavioral features: a) correctly-spelled words; b) emoticons, and; c) intervention words per user. The following subsections describe and analyze each one of them in detail.

5.5.1 Correctly-spelled words

Online conversations are characterized by the excessive use of misspelled words. In this context, the following research question can be postulated: “do groomer chats tend to have more mistyped words than non-groomer ones?”. Here, one may expect a positive answer, mainly because children employ a limited vocabulary due to their youth, and also because groomers commonly pretend to be children. But, there are other reasons for the presence of typos that make the answer unclear, such as poor education, lack of attention, laziness, or even the occurrence

Figure 18 – Elements of writing style present in groomer and non-groomer conversations from the SGD training set.



(a) Overall percentage of terms that are correctly-spelled words.

(b) Overall percentage of terms that are identified as emoticons.

(c) Overall percentage of terms written by the top-5 users that most intervene in each conversation.

of intentional errors made for fun or to prevent automatic analyses of the messages posted. To investigate this matter, we computed the percentage of correctly-spelled words in groomer and non-groomer chats from the SGD training set, where a term is considered to be a correctly-spelled word if and only if it belongs to the lexical database of English WordNet[4]. Figure 18a reports the corresponding results. As it can be seen, correctly-spelled words are more common in the groomer chats than they are in the non-groomer ones, which is the opposite of what we expected at first glance. Nevertheless, the percentage of correctly-spelled words offers considerable contrast between the classes. Therefore, we propose to take advantage of it as one of the behavioral features that describe a conversation in our framework.

5.5.2 Emoticons

In the past decades, emoticons have become important tools to transmit emotions and sentiments in online conversations. Therefore, we decided to investigate whether or not groomer and non-groomer chats present distinct patterns in the use of such tools. To make it possible, we took advantage of library SoMaJo tokenizer[5] to identify the many different character combinations that may compose one emoticon. As was expected, a wide variety of emoticons exist in the conversations from the SGD training set. Specifically, more than 350 distinct types of emoticons were identified in the corpus. Let us use Figure 19 to report the most popular ones considering each class separate. Interestingly, the emoticons that are popular for class *G* are considerably different than the ones of class *NG*. For example, the three most popular emoticons in groomer conversations are ":)" , ":-*" and ":(", while emoticons ":)" , ";)" and ":D" prevail in non-groomer conversations. However, it is not clear how to define behavioral features that take advantage of this characteristic, since the use of features that target specific emoticons would increase too much the sparsity and the dimensionality of the feature space. One can also note in Figure 19 that the overall counts of occurrences are much larger for class *NG* than they are for class *G*, but these differences alone may be meaningless because the dataset has considerably fewer groomer chats than non-groomer ones. To account for the data imbalance, Figure 18b reports the overall percentage of terms that are emoticons considering conversations of each class separate. It turns out that the percentage of emoticons is actually a bit higher for the groomer conversations. Note that there is little contrast between the classes since the percentages are relatively similar, but still this feature contributes to reveal groomer behavior as it is shown later in the experimental section; see Chapter 7. Therefore, we propose to take advantage of the percentage of terms that are emoticons as one of the behavioral features that describe a conversation in our framework.

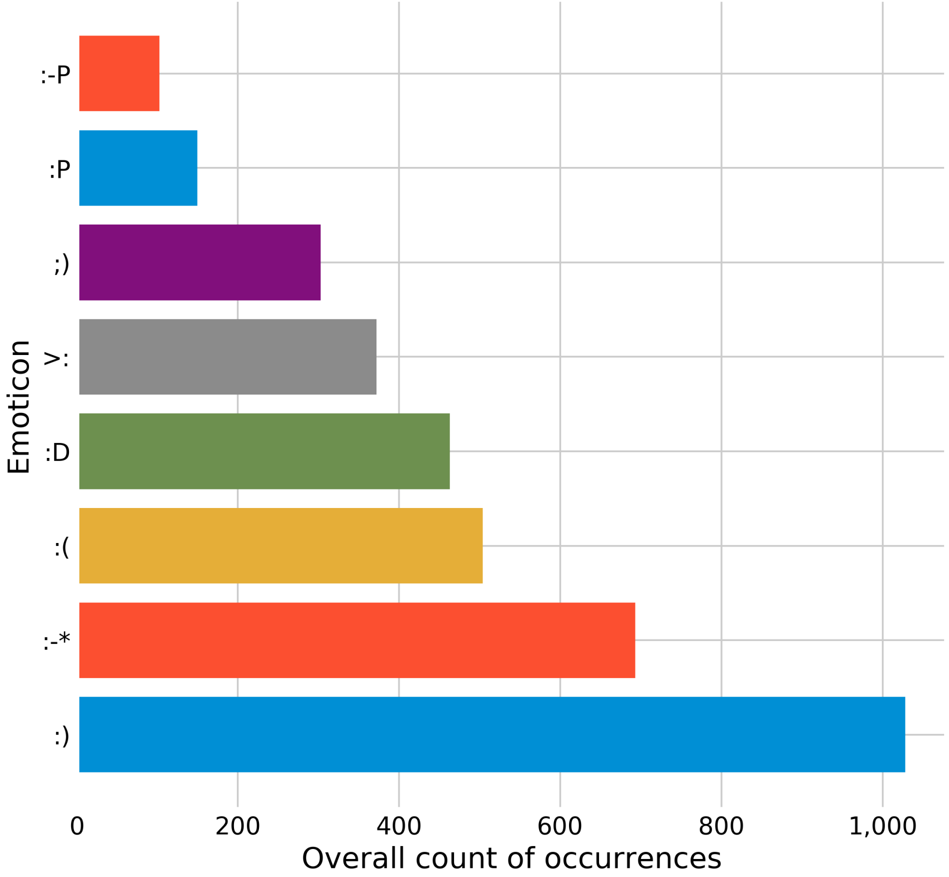
Finally, it is important to note that we intended to perform a similar analysis regarding the use of emojis, which are also largely employed to express emotions and sentiments in online conversations. Unfortunately, it was not possible because not a single emoji exists in the SGD

[4] <https://wordnet.princeton.edu>

[5] <https://github.com/tsproisl/>

Figure 19 – Most frequent emoticons in groomer and non-groomer conversations from the SGD training corpus.

(a) Groomer conversations.



(b) Non-groomer conversations.

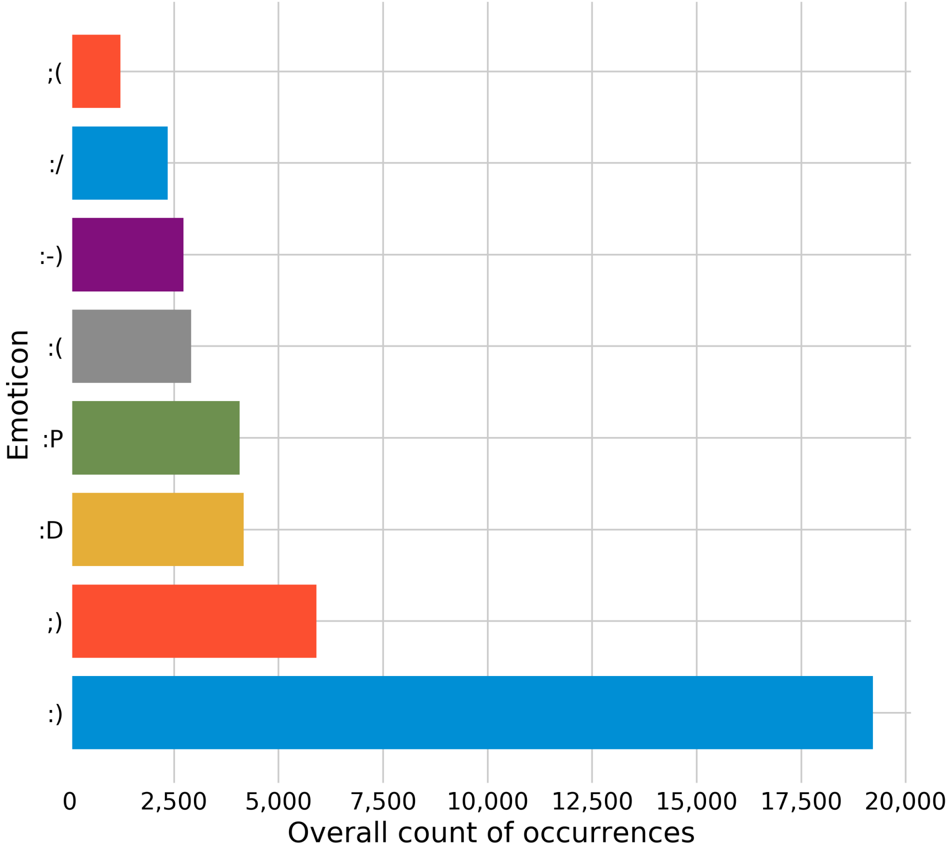
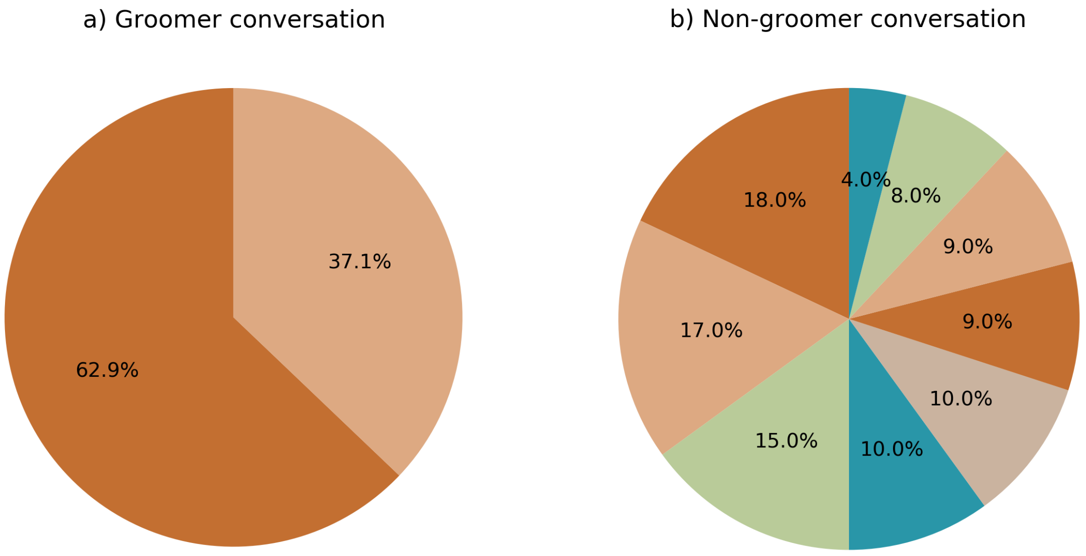


Figure 20 – Percentage of terms written by each user considering two random chats from the SGD training set. Note that the groomer chat has 2 participants, and one of them intervenes much more than the other. A distinct pattern is seen in the non-groomer chat, as it has many participants, and no one stands out with much more intervention words than the others.



training set, which is probably due to the antiquity of the dataset that was created in 2012. Nevertheless, we believe that there are strong similarities in the use of emoticons and emojis. Therefore, when processing datasets that include emojis, we suggest counting emojis together with emoticons when computing the behavioral feature proposed in this subsection.

5.5.3 Intervention words per user

The number of intervention words per user has never been employed before to reveal groomers. As the usage of correctly-spelled words and emoticons, the number of intervention words is considered to be one component of the writing style of users. It tends to reflect their interest in conversations, since those users that rarely intervene in a conversation may not be much interested in chatting, while other users that intervene constantly may be more interested in it. With that in mind, we investigated whether or not the patterns present in the interventions of users are distinct for groomer and non-groomer chats. Here, instead of limiting ourselves to counting the number of interventions from each user, we counted the number of words employed in the interventions, since there are cases where a user only intervenes a few times, but the content of each message is rich, while other users have many interventions with few words each. Thus, we considered the percentage of words written by a user with respect to the total number of words in each chat.

To illustrate the patterns behind the interventions of users, we randomly chose and analyzed one groomer and one non-groomer conversation from the SGD training set. Figure 20 summarizes our findings. As it can be seen, there are only two participants in the groomer conversation, and one of them intervenes much more than the other, thus being responsible for more than 60% of the words written. On the other hand, the non-groomer conversation has a large number of participants, and none of them stands out with much more intervention words than the others. Figure 18c provides the big picture by considering all conversations from the SGD training set. It reports the overall percentage of words written by the top-5 users that most intervene in each conversation, where the results for groomer and non-groomer conversations are shown separately. Here, it is important to mention that we considered only the top-5 users that intervened more in each conversation since they are the ones that provide the most valuable information. As it can be seen, the top-5 participants in the groomer chats interact considerably more than those in the non-groomer ones. It indicates that the participants of groomer conversations tend to use more words to communicate than those of non-groomer chats, which may reflect one large interest of groomers towards the victims. Provided that the number of intervention words per user offers considerable contrast between the classes, we propose to take advantage of this fact by employing the percentage of words written by each user with respect to the total number of words as one of the behavioral features that describe a conversation in our framework. Note that only the top-5 participants are considered to ensure that the same number of features is extracted per conversation; chats with less than five participants have a value of zero for some of the features.

5.6 Summary and discussion

This chapter presented the behavioral features that describe a conversation in our framework. To tackle the lack of analysis seen in the works of the forensic scenario, we studied each feature in detail aimed at understanding the reasons why it is helpful to reveal grooming. Here, it is worth mentioning that we also considered the use of other features that are popular in the forensic scenario, e.g., link exchange (URLs), mentions (@user), hashtags (#), number of messages, use of punctuation, etc. However, they did not provide considerable gains in the scenario of early detection. The following chapter presents our second major contribution: the BF-PSR framework for the early detection of grooming in online conversations.

CHAPTER 6

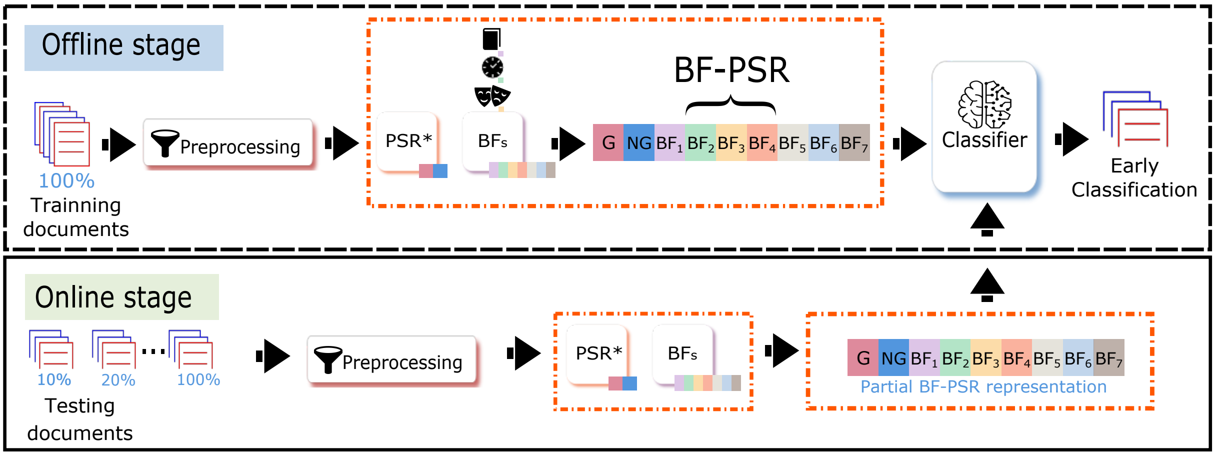
PROPOSED BF-PSR FRAMEWORK

As the second major contribution of our work, this chapter introduces the new framework Behavioral Feature – Profile Specific Representation (BF-PSR). It is an extension of the existing PSR framework in such a way that we adequately insert our proposed behavioral features in a vector space, which remains non-sparse and low-dimensional as it is desirable, but now contains more valuable patterns that highlight the differences between groomer and non-groomer conversations. To achieve this contribution, some fundamental adjustments were made to the original PSR vector representation, which we name as PSR*. By combining our seven behavioral features with PSR*, the BF-PSR vector representation is obtained as our formal proposal. Figure 21 illustrates the pipeline of our framework. It is divided into two main stages:

i) **Offline stage:** The offline stage is dedicated to the analysis of the documents available for training. The first step is the use of preprocessing techniques to remove noise from the data. Then, fundamental modifications are performed to the original PSR method which we name as PSR*. With this improved method, the preprocessed documents are represented in a distinct vector space that remains non-sparse and low-dimensional. Simultaneously, a novel vector representation is obtained from the behavioral features proposed in the previous Chapter 5. Later, both representations are concatenated into a single vector representation space named BF-PSR, which is used to train the classifier.

ii) **Online stage:** In the online stage, the testing documents are divided into ten portions that contain different percentages of the information available, i.e., 10%, 20%, ... and 100% so that we simulate the sequential arrival of messages in online conversations. Then, each portion of the text is preprocessed and the BF-PSR vectors are extracted from the available information. Let us name these vectors as partial BF-PSR representations since they consider only the available information, not the entire conversations. At last, the partial representations are used to perform the early classification.

Figure 21 – Two stages describe the pipeline of our proposal. **Offline stage**: The training conversations are preprocessed and each one is represented by a PSR* vector. Then, BFs are computed to highlight the behavioral patterns occurring in the different classes, i.e., groomer and non-groomer conversations, and to represent each conversation in a novel vector. Both vectors, PSR* and BFs, are then concatenated into a single vector named BF-PSR. Later, the classifier is trained with this new document representation space. **Online stage**: The testing set is divided into ten portions containing different percentages of the information available, i.e., 10%, 20%, ..., and 100%. Then, the portions of information are preprocessed and represented by partial BF-PSR vectors. Finally, the classifier performs predictions with the early data provided. As a side note, the parts of the pipeline highlighted by rectangles with red borders are the contributions of our proposal that essentially differentiate BF-PSR from the original PSR framework.



The sections in the following describe each of the aforementioned stages in detail.

6.1 How to preprocess the corpus

The first step in our pipeline is to preprocess the input dataset. In the case of the SGD corpus, as described in the previous Chapter 3, it was necessary to perform a relabeling process where all conversations with a groomer participant were cataloged as *G*-conversations, while the others were considered to be *NG*-conversations. Besides, in both the training and the testing sets, a chat-level document representation was applied so to represent each conversation as a document. Next, we empirically tested different combinations of the preprocessing techniques described in Chapter 2. Recall that knowing which techniques to apply and in which order depends on the dataset and the classification task. For our specific case, it was only necessary to apply the combination of two of the described techniques, i.e., removing XML attributes and lowercasing the tokens. Thus, the SoMaJo tokenizer[1] was employed to remove XML elements, like entities and tags, besides empty conversations. And then, the text was transformed to lowercase where the tokens were maintained as uni-grams. Note that, the most common preprocessing techniques

[1] <https://github.com/tsproisl/>

in the literature were not applied such as removing stopwords (e.g., “are”, “is”, “the”), replacing abbreviations (e.g., “omg”) and punctuation (e.g, “!!!!”), removing emoticons(e.g., “:)”,“:-*”), etc. This is because the more tokens were removed from conversations, the more information was lost. Confirming that maintain the writing style of the users improves the performance in the early detection of grooming. Additionally, let us highlight that **no filtering** process was applied when preprocessing the data. As it was discussed in Section 4.2.4, we believe that the creation of filtering rules, such as ignoring conversations where only one user participates or those with few interactions, is not suitable in real-life systems that must detect a potential risk. The truth is that many of these chats can be explicit attempts of groomers to contact their victims, so they cannot be ignored. Besides, in our humble opinion, we understand that the state-of-the-art methods that apply filtering techniques do not present accuracy results that could be replicated in a real system. This is because they commonly disregard the errors coming from groomer chats that were mistakenly filtered out of the analysis. In our work, the complete information of the training documents is used, and no groomer conversation of the entire corpus, i.e., training and testing documents, is lost due to the preprocessing procedure.

6.2 How to compute and use our proposed PSR* vector representation

In our proposal, we take advantage of the PSR method to represent the conversations in a non-sparse and low-dimensional vector space. However, it is important to note that the original PSR was evaluated only with filtered data, and, when we tested it with unfiltered data, the results were unfortunately disappointing. These results are reported in Figure 1 from our introductory section. Therefore, to keep representing the documents in a non-sparse and low-dimensional vector space, some fundamental adjustments to PSR were necessary. First, the original preprocessing type was replaced by the one described in Section 8.1. Distinctly from the original PSR, we do not use character 3-grams to represent the tokens of the vocabulary; instead, we use regular uni-grams. Besides, one of the most critical improvements that we propose is to keep the emoticons and the punctuation as regular uni-grams. This modification truly improved the performance of our method and confirmed, once again, that maintaining the writing style of the users is essential to the early detection of grooming. Another important adjustment was performed in the minimum frequency of terms. When building the document-term matrix, the original PSR ignores terms with frequencies lower than 5 in the training set. Here, we propose to ignore terms with frequencies lower than 3 so to have more terms in the vocabulary. It highly reduces the “unknown term” detection. We also decided to replace the document-matrix structure with dictionaries to avoid high memory costs when the matrix is sparse. Finally, after performing a series of experiments with different types of term-weighting schemes, e.g., Term Frequency Inverse Gravity Moment (TF-IGM), Term Frequency Chi-square (TFχ^2), Term Frequency Relevance Frequency (TF-RF) and Term Frequency Probability-based (TF-Prob), the original

Algorithm 1 – Computing a behavioral feature that is associated with one or more lists of terms

```
Require:
    $documents$: the input conversations;
    $lists$: the lists of terms, where $lists[i]$ is the $i^{th}$ list of terms;
    $num\_lists$: the total number of lists;
Ensure:
    $bf$: matrix with the feature values computed. It has one value per list, per document;

Initialize $bf$ as a matrix with $|documents|$ lines and $num\_lists$ columns;
$id = 1$;
for each $document$ in $documents$ do
    Tokenize $document$ into $tokens$;
    for $i = 1, 2, \ldots num\_lists$ do
        $counter = 0$;
        for each $token$ in $tokens$ do
            if $token$ exists in $lists[i]$ then
                $counter{+}{+}$;
            end if
        end for
        $bf[id][i] = counter/|tokens|$;
    end for
    $id = id + 1$;
end for
return $bf$;
```

Term Frequency (TF) value used in PSR was replaced with the Term Frequency times Inverse Document Frequency (TF.IDF) in Equation 4.6. The use of TF.IDF led to notably better results, especially when considering the weights given to terms that are closely related to the groomer class. Therefore, we have named as PSR* the method that contains all of the aforementioned modifications. In the pipeline of Figure 21, the groomer and the non-groomer profiles of PSR* are respectively represented by the squares 🟥 and 🟦. Later, in the experimental section in Chapter 7 it is shown that as a result of having applied this set of configurations to the original PSR, the contribution produced allows overcoming the problem of the data imbalance in a very effective way. Thus, PSR* is one of our major contributions to the area of early detection of grooming in online conversations.

6.3 How to compute and use our proposed behavioral features

The previous Chapter 5 presented a detailed analysis of each of the behavioral features that we propose to employ in the early detection of grooming. This section explains how to compute and use these features in our framework. The seven BFs are represented in the pipeline of Figure 21 as the squares 🟪, 🟩, 🟨, 🟧, ⬜, 🟦, and 🟫. Let us begin with the features that are associated with one

Algorithm 2 – Composing the BF-PSR vector representation

```
Require:
    $num\_docs$: the number of input conversations;
    $g\_profile$: PSR* profile for class groomer. It is a $num\_docs \times 1$ matrix of real values;
    $ng\_profile$: PSR* profile for class non-groomer. It is a $num\_docs \times 1$ matrix of real values;
    $bf_1$: BF number of participants in a conversation. It is a $num\_docs \times 1$ matrix of real values;
    $bf_2$: BF sentiment and emotional markers. It is a $num\_docs \times 10$ matrix of real values;
    $bf_3$: BF sexual topic words. It is a $num\_docs \times 1$ matrix of real values;
    $bf_4$: BF time when a conversation starts. It is a $num\_docs \times 2$ matrix of real values;
    $bf_5$: BF correctly-spelled words. It is a $num\_docs \times 1$ matrix of real values;
    $bf_6$: BF emoticons. It is a $num\_docs \times 1$ matrix of real values;
    $bf_7$: BF intervention words per user. It is a $num\_docs \times 5$ matrix of real values;
Ensure:
    $bf\_psr$: the proposed BF-PSR vector representation. It is a $num\_docs \times 23$ matrix of real values;

$bf\_psr = [g\_profile\ \ ng\_profile\ \ bf_1\ \ bf_2\ \ bf_3\ \ bf_4\ \ bf_5\ \ bf_6\ \ bf_7]$;
return $bf\_psr$;
```

or more lists of terms, that is: a) the feature of **sentiment and emotional markers**, which is associated with one list of terms per emotion or sentiment; b) the feature of **sexual topic words**, which is associated with one list of sexual terms; c) the feature of **correctly-spelled words**, which is associated with one list of English words, and; d) the feature of **emoticons**, which is associated with one list of emoticons. Algorithm 1 is the pseudo-code that computes these behavioral features. It calculates the percentage of terms from each document that belong to each list and stores the values in a vector space. More specifically, the algorithm receives as parameters the set of documents to be processed, i.e., *documents*, the lists of terms to be considered, i.e., *lists*, where $lists[i]$ refers to the i^{th} list, and the corresponding number of lists, i.e., *num_lists*. The contents of parameters *lists* and *num_lists* vary depending on the feature to be processed. For example, when calculating the ratios of the sentiment and emotional markers, parameter *lists* contains the NRC and the DepecheMood lexicons with each list being related to one of the ten emotions or sentiments that we consider, i.e., anger, anticipation, joy, positive, sadness, surprise, amused, angry, inspired, and sad; and, *num_lists* is ten, so a total of ten columns compose this feature. Meanwhile, when calculating the ratio of the sexual topic words, parameter *lists* has a single list, i.e., the extended list of sexual terms from Section 5.3, so $num\_lists = 1$ and only one column is necessary to compose this feature. Having these examples in mind, in Line 1, the algorithm allocates memory for one matrix named *bf* that is used to store the feature values. Then, in Lines 3-11, each document suffers a tokenization process to identify the presence of any term that belongs to any of the lists. In Line 10, the ratio of terms from a document that is associated with one of the lists is calculated with respect to the total number of terms in the document. Once all documents have been processed, the corresponding feature values are returned in Line 12. The computation of the remaining BFs is as follows:

- **Number of participants in a conversation:** in this case, matrix bf has $|documents|$ lines and one single column to store the number of unique users participating in each conversation. Note that, in the online stage of our framework, the values of this feature may change according to the information available, as the participants are counted only after they post their first message.

- **Time when a conversation starts:** in this case, matrix bf has $|documents|$ lines and two columns with the hour (HH) and the minute (MM) of the first message sent in each conversation. Note that each column of this BF is min-max normalized to maintain the homogeneity in the final vector. Unlike the other behavioral features, these columns remain static in the online stage, regardless of the amount of information available. Here, we only care about the time when each conversation started.

- **Intervention words per user:** in this case, matrix bf has $|documents|$ lines and 5 columns because we register only the information of the five users that participate the most in each conversation. Then, we compute the ratio of the number of words written by each user with respect to the total number of words in the conversation. The first column of matrix bf stores the ratio of the participant with more interactions, i.e., the one of the most interactive user; the second column has the ratio of the second most interactive user, and so on. This process is repeated for each conversation; chats with less than five participants have a value of zero for some of the columns.

Table 16 summarizes the construction of each behavioral feature and its corresponding numbers of columns. Once we concatenate the PSR* representation, from the previous Section 6.2, with the bf matrix of each of our seven behavioral features, there are 23 columns in total; see the details in Algorithm 2. This is the final representation that we propose to use for the early detection of grooming, that is, the new BF-PSR vector representation. Note that this new representation is still non-sparse and low-dimensional, as it is desirable, but now it contains more valuable patterns that highlight the differences between groomer and non-groomer conversations. Finally, let us emphasize that the user is free to choose any classifier to employ in our framework. Nevertheless, we performed an extensive evaluation of many of the state-of-the-art classifiers, and classifier MultiLayer Perceptron obtained the best results. Therefore, we suggest its use in the BF-PSR framework. The details of the aforementioned evaluation, as well as the corresponding results, are given later in this monography; see the experimental section in Chapter 7.

6.4 How to perform the early detection of grooming

Once the classifier is trained in the offline stage of our framework, the sequential arrival of messages is simulated in the online stage to perform and evaluate the early text classification. At first, the testing corpus is divided into ten portions containing 10%, 20%, ... and 100% of each

Table 16 – Summary of how each of our seven behavioral features is computed for one conversation. We report the name of the feature, a brief description of how it is computed, and the corresponding number of columns that compose the feature.

Behavioral feature	How to compute it for one conversation	Number of columns
Number of participants in a conversation	Count the distinct users that posted one or more messages up to the moment of analysis.	One column composes this behavioral feature.
Sentiment and emotional markers	For each emotion considered: $\left\{\frac{\text{\# of terms related to the emotion}}{\text{Total number of terms}}\right\}$	Ten columns compose this behavioral feature, one per emotion: anger, anticipation, joy, positive, sadness, surprise, amused, angry, inspired and sad.
Sexual topic words	$\left\{\frac{\text{\# of terms related to the sexual topic}}{\text{Total number of terms}}\right\}$	One column composes this behavioral feature.
Time when a conversation starts	Identify the time (HH:MM) when the first message was posted.	Two columns compose this feature, one for the hour (HH) and the other for the minute (MM).
Correctly-spelled words	$\left\{\frac{\text{\# of terms within the English lexicon}}{\text{Total number of terms}}\right\}$	One column composes this behavioral feature.
Emoticons	$\left\{\frac{\text{\# of terms identified as emoticons}}{\text{Total number of terms}}\right\}$	One column composes this behavioral feature.
Intervention words per user	For each of the top-5 users: $\left\{\frac{\text{\# of terms written by the user}}{\text{Total number of terms}}\right\}$	Five columns compose this behavioral feature, one for each of the five most participative users.

conversation. More specifically, we calculate the total number of tokens of each conversation of the testing set. Then, to simulate a conversation where only 10% of the information is available, a new testing subset is created. Here, only the messages in chronological order whose total number of tokens do not exceed the 10% of the whole conversation tokens are considered. This process is repeated for all the text portions to be simulated, i.e., 10%, 20%, ... and 100%. Then, the same preprocessing and feature extraction performed in the offline stage is reused in the online stage, in such a way that we obtain partial BF-PSR vector representations for each conversation. Finally, each partial BF-PSR vector is given, one by one, to the classifier to perform the early predictions.

6.5 Summary and discussion

This chapter presented the new framework BF-PSR for the early detection of grooming. In the next chapter, we validate our proposals by reporting the results of an extensive experimental evaluation; they indicate that the new framework achieves state-of-the-art accuracy when processing the full SGD corpus.

CHAPTER 7

RESULTS AND DISCUSSION

Results and discussion This chapter describes and reports the results of an extensive experimental evaluation performed to validate the new BF-PSR framework. Specifically, we aimed at answering the questions in the following.

Q1 Compared with five of the best and recent previous approaches in a scenario where behavioral features are disregarded, how effective is PSR*?

Q2 How much do the proposed behavioral features contribute to distinguish the classes?

Q3 How effective is our final proposal BF-PSR compared both with PSR* and also with five of the best and recent previous approaches?

Q4 How effective is our BF-PSR framework when evaluated with a dataset different from the SGD testing corpus?

The experimental evaluation was performed in a machine with 32GB of RAM and a processor CPU Intel Xeon working at 2.6GHz, under OS Ubuntu 18. BF-PSR and PSR* were compared with five of the best and recent previous algorithms, that is, the ENB, the original PSR, and the three variants of MulR, namely, MulR-W2V, MulR-PSR, and MulR-TVT. Here, it is important to highlight that we requested the original authors of the previous methods for their implementation. The authors of both PSR and MulR kindly responded to our request. We thank them very much for that. Unfortunately, the original MATLAB code of PSR produced a "memory exceeded error" when processing the complete SGD corpus, although it works flawlessly in the filtered SGD corpus. We believe that it happens due to memory limitations that are intrinsic to MATLAB. On the other hand, the code provided by the authors of MulR corresponded to an unpublished variant of their method, which is none of the ones described in their paper; so, it was only possible to avail of certain portions of the original code. Finally, we were unable to obtain the ENB method code, despite contacting the authors. Due to the aforementioned reasons, we implemented in

Python all of the algorithms studied. The correctness of our reimplementation for the previous methods was verified by comparing the results obtained when processing the filtered SGD corpus. Specifically, because we did not have access to the original code, we compared our Python version of the ENB with the corresponding results reported in the original paper. In the case of PSR, we compared our Python version with the original source codes in MATLAB. And for MulR, provided that we also could not access the original source codes of the variants that are described in the publication, we compared the three variants that we implemented in Python with the corresponding results reported in the original paper.

Reproducibility: for the purpose of reproducibility, all codes, including the Python implementation for the state-of-the-art methods and for our own methods, besides the detailed results obtained, the parameter values tested and the data studied in our paper, e.g., PJZ and PJZC datasets. Are freely available for download online[1].

7.1 Evaluating effectiveness without behavioral features

This section investigates Question **Q1** by comparing the effectiveness of five of the best and recent state-of-the-art methods with that of our PSR* when processing the SGD corpus. Here, the focus is to evaluate the improvements that we propose for the original PSR; thus, it is considered a scenario where behavioral features are disregarded.

We first verified the correctness of our Python implementation of the previous methods. As it was described before, the correctness of our reimplementations was verified by comparing the results returned from our code with those ones provided by the original authors of each previous work, considering the filtered SGD corpus. Thus, filtering techniques were applied to the dataset.

For ENB, the original authors filtered out of the analysis conversations with only one participant, and those that have less than 6 interventions per user, besides long sequences of unrecognized characters that were apparently images. We replicated the very same procedure for this particular experiment. As a result of the filtering process, 856 (10.33% of the filtered SGD training set) groomer conversations and 7,428 (89.67% of the filtered SGD training set) non-groomer conversations were obtained for the SGD training set. While the SGD testing set yielded 1,573 (8.2% of the filtered SGD testing set) groomer conversations and 17,602 (91.80% of the filtered SGD testing set) non-groomer conversations. Note that as a consequence of the filtering process, almost 60% of the groomer conversations in the training and testing set were removed. Following, Figure 22a reports, with a red dotted line, the $F1_g$ results reported in the original paper with different percentages of information available. In turn, the brown dotted line represents the $F1_g$ results of our own implementation. Note how with 30% of the available information, the curves begin to be similar and with 80% of the available information, our reimplementation begins to exceed the accuracy of the original paper. On the other hand, it seems

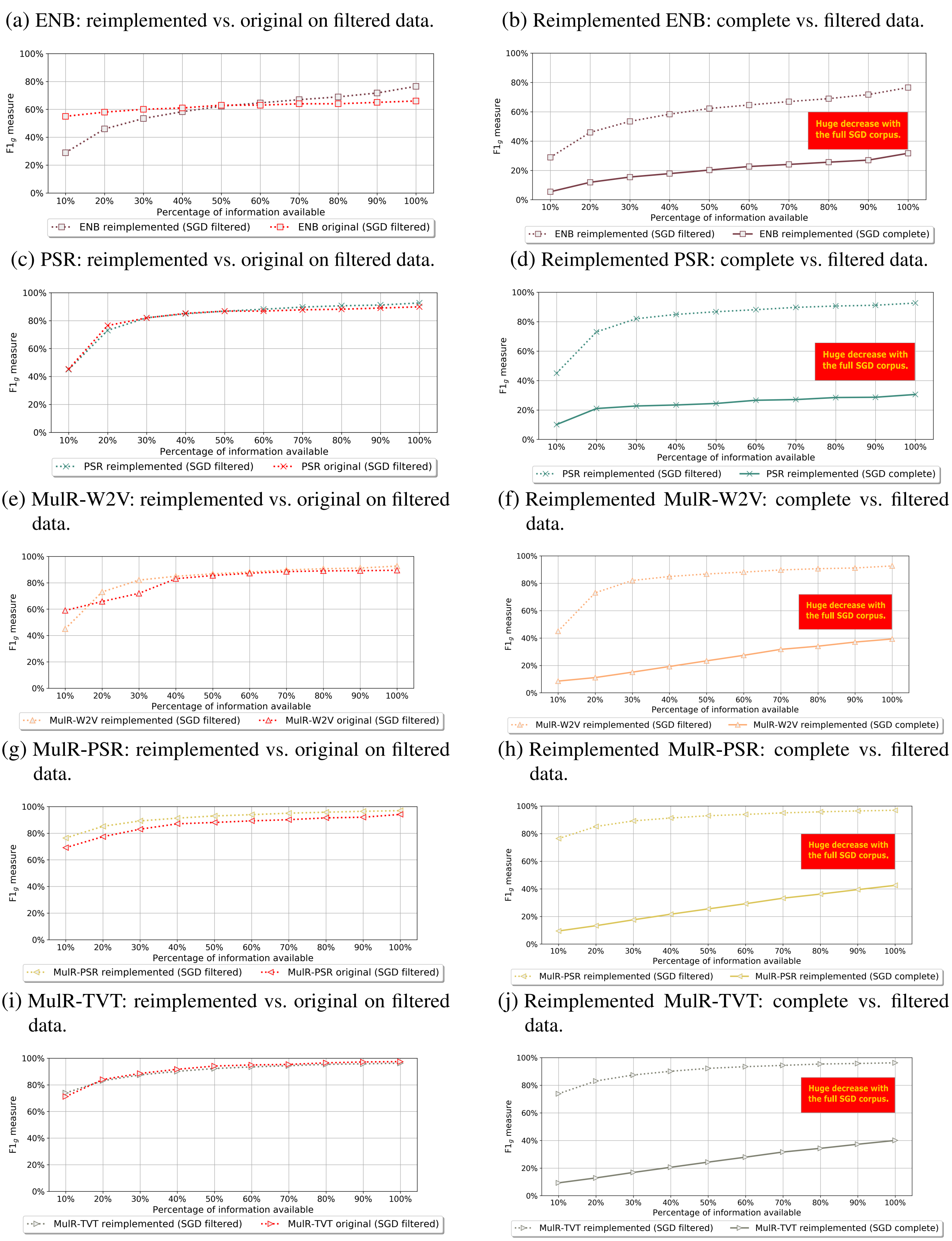
[1] <https://github.com/danielafe7-usp/BF-PSR-Framework>

that our reimplementation struggles to reach the accuracy of the original paper with 10% and 20% of the available information. We did our best to extract as much detail as possible from the original paper and implement the method as accurately as possible. Still, our reimplementation returned results that are similar to the original ones in the remaining portions of text available, so we consider it to be successful.

When analyzing the aforementioned results, one can see how ENB apparently reaches 60% in the $F1_g$ measure with 50% of the information available. However, these results **completely disregard** the groomer conversations that were filtered out by the system, **before** the analysis. In our humble opinion, the effectiveness of the method is being masked in this plot. Thus, to find out how effective the ENB method would be in a real-world application, we tested our reimplementation with the entire SGD corpus. That is, without performing any filtering process. Figure 22b reports the results of our reimplementation with the filtered (dotted line) and the complete (solid line) SGD corpus. From the plot, one can observe a huge decay in effectiveness when the complete corpus is analyzed; in this case, when 50% of the information is available, ENB barely reaches 20% in the $F1_g$ measure, and it improves only to nearly 32% $F1_g$ with more information. These results highlight the ineffectiveness of ENB to process large amounts of disproportionate data. Also, as a brief observation to continue with this subsection, the dotted lines in the plots of Figure 22 always represent results obtained with the filtered SGD corpus; the solid lines refer to those ones obtained from the complete dataset.

Now, let's analyze the PSR method. The authors of PSR used the same filtering process proposed before for ENB, so we applied the same procedure for our reimplementation. Then, Figure 22c reports the $F1_g$ results achieved by the original code and our implementation with different percentages of information available. We can observe the similitude of both curves in the plot which easily leads us to conclude that our reimplementation of PSR was successful. Then, when analyzing the results, PSR reaches 75% of $F1_g$ measure with only 20% of the information available and about 95% of $F1_g$ measure when all the information is available. Once more, let us emphasize that the aforementioned results **completely disregard** the groomer conversations that were filtered out by the system, **before** the analysis, so we argue that the effectiveness of PSR is being masked in this plot. With this in mind, let's see how PSR actually performs when the complete SGD corpus is analyzed. From Figure 22e, we can observe that the PSR method presents a huge decay in effectiveness. Thus, when only 20% of the information is available, the method barely reaches 20% in the $F1_g$ measure, and it improves only to nearly 30% $F1_g$ when 100% of the information is available. Therefore, this method also struggles when there is a large amount of unbalanced data. As an additional note, remember that we intended to perform the same experiment with the original code in MATLAB, but, due to intrinsic characteristics of this program, the code produces a "memory exceeded error" with the complete corpus.

Finally, let us focus on MulR. We performed a similar analysis for its three variants MulR-W2V, MulR-PSR, and MulR-TVT. As it happens with ENB and PSR, the authors of MulR also



(a) ENB: reimplemented vs. original on filtered data.

(b) Reimplemented ENB: complete vs. filtered data.

(c) PSR: reimplemented vs. original on filtered data.

(d) Reimplemented PSR: complete vs. filtered data.

(e) MulR-W2V: reimplemented vs. original on filtered data.

(f) Reimplemented MulR-W2V: complete vs. filtered data.

(g) MulR-PSR: reimplemented vs. original on filtered data.

(h) Reimplemented MulR-PSR: complete vs. filtered data.

(i) MulR-TVT: reimplemented vs. original on filtered data.

(j) Reimplemented MulR-TVT: complete vs. filtered data.

$F1_g$ measure
0%
20%
40%
60%
80%
100%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Percentage of information available
ENB reimplemented (SGD filtered)
ENB original (SGD filtered)
ENB reimplemented (SGD complete)
PSR reimplemented (SGD filtered)
PSR original (SGD filtered)
PSR reimplemented (SGD complete)
MulR-W2V reimplemented (SGD filtered)
MulR-W2V original (SGD filtered)
MulR-W2V reimplemented (SGD complete)
MulR-PSR reimplemented (SGD filtered)
MulR-PSR original (SGD filtered)
MulR-PSR reimplemented (SGD complete)
MulR-TVT reimplemented (SGD filtered)
MulR-TVT original (SGD filtered)
MulR-TVT reimplemented (SGD complete)
Huge decrease with the full SGD corpus.

Figure 22 – Effectiveness of previous methods with the filtered (dotted lines) and the complete (solid lines) SGD corpus.

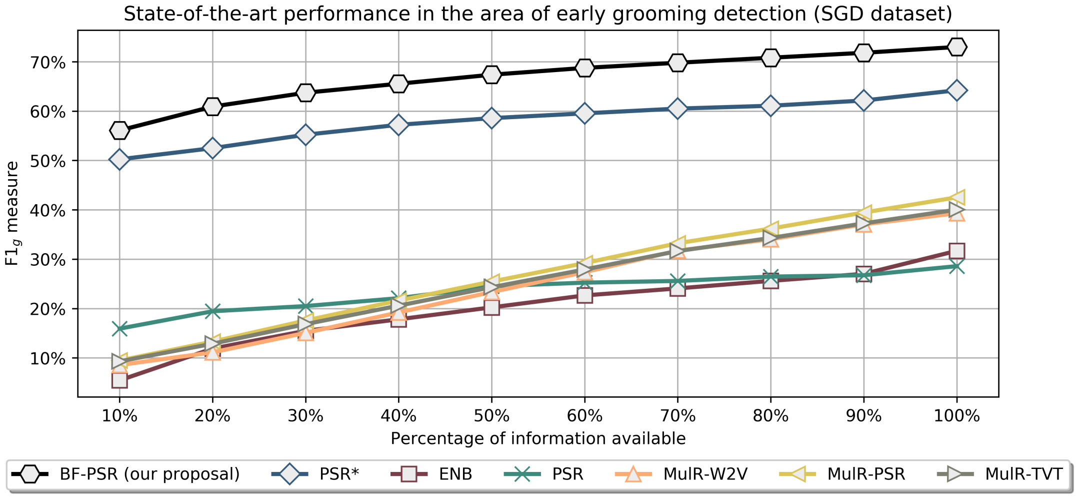


Figure 23 – Our framework BF-PSR is compared against PSR* and five previous methods in the area of early detection of grooming. The plot reports the accuracy of results achieved by each method when different percentages of text are available for processing. As it can be seen, methods ENB, PSR, MulR-W2V, MulR-PSR, and MulR-TVT fail to provide accurate results when processing the full SGD corpus. However, both of our approaches BF-PSR and PSR* provide much better results by being able to properly deal with the data imbalance, which is expected to exist in a real system. They both obtain F1$_g$ values that are larger than 50% when only 10% of the information is available, and they also improve the accuracy when having access to more information. As it is expected, our behavioral features allow BF-PSR to overcome PSR*; the gain in effectiveness is nearly 10% F1$_g$ through every portion of the information available. Thus, our final proposal BF-PSR obtains state-of-the-art performance in the early detection of grooming.

employed filtering techniques to the dataset. However, the original paper fails to specify the type and the details of the filtering procedure employed. Therefore, when trying to reproduce the original results of MulR, we reused the same filtered dataset created before to evaluate ENB. Figures 22d, 22g and 22i compare the results obtained by each variant of MulR, as reported in the original paper, with those obtained from the corresponding reimplementations in Python. As it can be seen, in Figure 22d, our reimplementation of MulR-W2V outperformed its original version when 20% or 30% of the information was available. We believe that this is because the method uses word embeddings that, over the years, have been improving and providing more information. Despite this fact, our reimplementation returned results that are similar to the original ones in the remaining portions of text available, so we consider it to be successful.

Continuing with the analysis, our reimplementation of MulR-PSR is depicted in Figure 22g. From the plot, it is possible to observe how our reimplementation is consistently better in each portion of the information available. We hypothesize that the reason for this difference is the unknown type of filtering process used in the original paper, which may differ from the one that we applied in our experiments. However, note that the relative difference in F1$_g$ values obtained from both implementations remains stable for nearly all portions of information available, so we consider that the reimplementation was also successful in this case.

Finally, as it can be seen in Figure 22i, our reimplemented MulR-TVT reported barely the same results as the ones reported in the original paper, so we also consider it to be a successful reimplementation.

One more time, let us emphasize that the aforementioned results **completely disregard** the groomer conversations that were filtered out by the system, **before** the analysis, so we argue that the effectiveness of MulR is being masked in the plots of Figures 22d, 22g and 22i. To find out how effective MulR would be in a real-world application, we tested our reimplementations on the entire SGD corpus. That is, without performing any filtering process. Figures 22f, 22h and 22j report the corresponding results with the $F1_g$ values obtained from our reimplementation of each variant of MulR when processing the filtered (dotted line) and the complete (solid line) SGD corpus. From the plots, one can observe a huge decay in effectiveness when the complete corpus is analyzed; in this case, when 20% of the information is available, all variants of MulR reach nearly 10% in the $F1_g$ measure, and they improve only to nearly 40% $F1_g$ with more information. These results highlight that, as it happens with ENB and PSR, method MulR is also ineffective when processing large amounts of disproportionate data.

Up to this point, we have been able to demonstrate the ineffectiveness of the previous methods when processing a large amount of unbalanced data, which is the expected scenario in a real system. Now, we evaluate how our PSR* behaves in this scenario. In other words, we evaluate our contributions to the original PSR when processing unfiltered data.

Figure 23 reports the corresponding results for PSR*, ENB, PSR, MulR-W2V, MulR-PSR, and MulR-TVT. Note that it is a detailed version of the motivational figure that we presented in the introductory section, i.e., Figure 1; so, Figure 23 also reports the results of our final proposal BF-PSR that are discussed later in the paper. As it can be seen, our PSR* clearly surpasses all the previous methods in effectiveness. It offers a gain of more than 30% $F1_g$ over the best competitor when only 10% of the information is available, and it maintains very large gains when having access to more information. These results demonstrate that the modifications proposed for the original PSR were fundamental to overcome the data imbalance. Due to this reason, we consider PSR* to be one of our major contributions in this work.

7.2 Evaluating the contribution of our proposed behavioral features

This section answers Question **Q2** by investigating whether or not each of our seven behavioral features contributes to the early detection of online grooming. So far we have concluded that PSR* is the best approach in a scenario that disregards behavioral features. Therefore, PSR* is compared in this subsection with alternative methods that build upon it and take advantage of each one of our behavioral features individually. More specifically, we created seven alternative vector representations by concatenating the profiles *g_profile* and *ng_profile* of PSR* with the matrix bf of each individual behavioral feature. Then, the alternative vector representations were compared with the plain PSR* representation so to understand how much each feature contributes to highlighting groomer behavior. The well-known classifiers in the following were evaluated to perform the early predictions: MultiLayer Perceptron (MLP), Gradient Boosting Machines (GBM), *k*-Nearest Neighbors (*k*NN), XGBoost (XGB) and Random Forest (RF). Table 17 reports the best classifier identified – and, thus used – for each of the vector representations studied in this subsection.

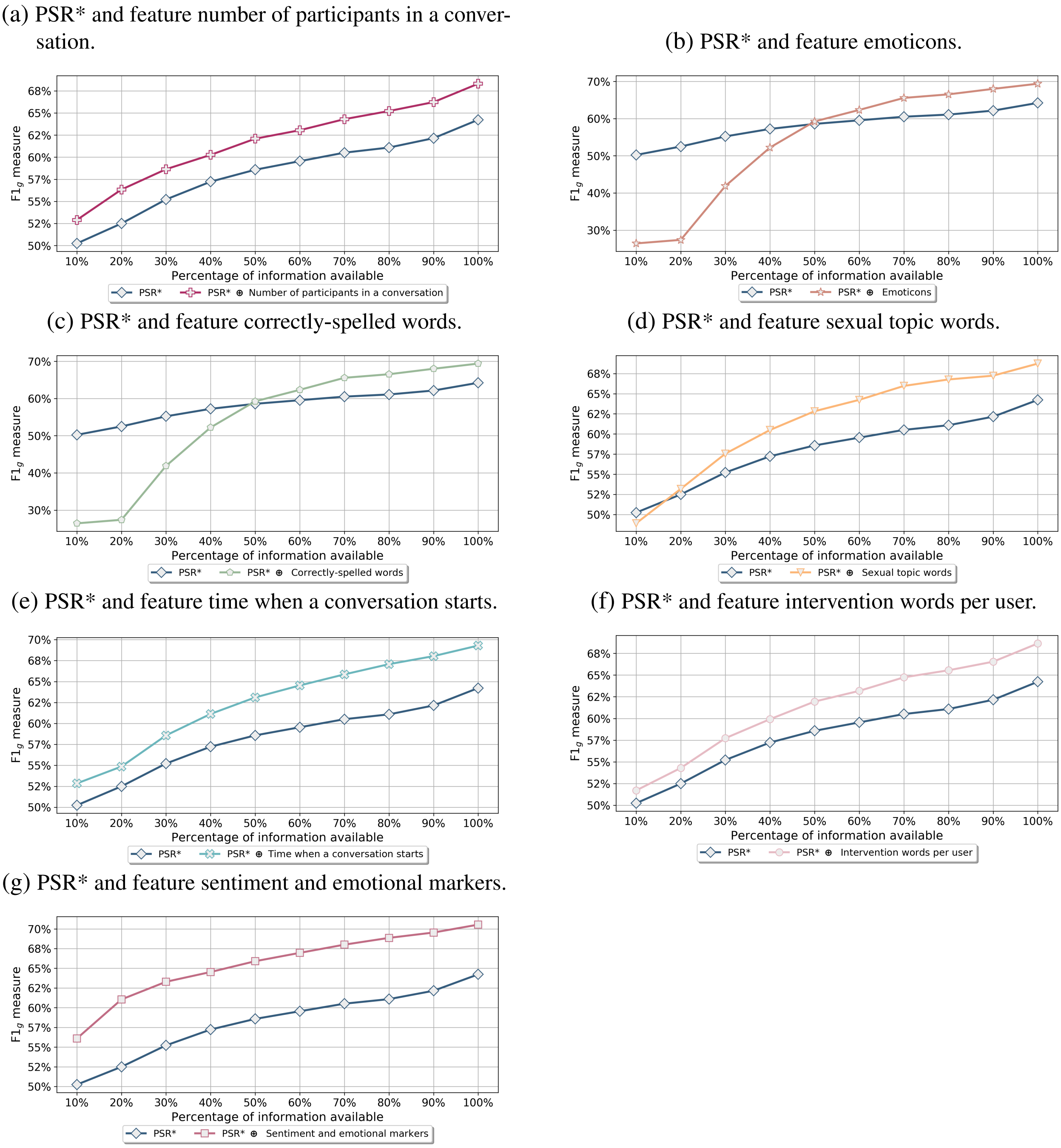
Vector representation	Best classifier
PSR* and feature number of participants in a conversation	GBM
PSR* and feature emoticons	GBM
PSR* and feature correctly-spelled words	GBM
PSR* and feature sexual topic words	XGB
PSR* and feature time when a conversation starts	GBM
PSR* and feature intervention words per user	XGB
PSR* and feature sentiment and emotional markers	*k*NN
BF-PSR, that is, PSR* and all seven BFs combined	MLP

Table 17 – Best classifiers were identified and used for each of the alternative vector representations that we studied. Classifier MultiLayer Perceptron (MLP) obtained the best results from our final proposal, the BF-PSR vector representation.

The results of the aforementioned analysis are reported in Figure 24. The first feature that we evaluated was the number of participants in a conversation. Figure 24a shows the performance results achieved by the alternative vector representation that includes this feature against the plain PSR* representation. As it can be seen, the use of this feature leads to a considerable gain in accuracy. More specifically, the gain occurs along all the portions of information available, including the early portions that contain the least amount of information.

Continuing with the analysis, one can observe from Figures 24b and 24c that features emoticons and correctly-spelled words offer a smaller contribution to the online detection of grooming. More specifically, it can be observed that feature emoticons contributes only when half or more of the total information is available, i.e., 50%, 60%, ... 100%. When there is few information to be analyzed, for example, 10%, the performance of the method decreases severely. This pattern is repeated with feature correctly-spelled words. We believe that it occurs because both of these features offer low contrast between the classes groomer and non-groomer, as it was discussed before in Subsections 5.5.1 and 5.5.2; see Figures 18a and 18b for details. Despite this fact, when combined with our other five BFs, both features correctly-spelled words and emoticons become valuable, as they do not harm accuracy when few information is available, and still increase accuracy when having more information to analyze.

The remaining plots indicate that all of the other BFs offer notably large gains in accuracy for all the portions of information considered. The only exception is a small decrease in accuracy



(a) PSR* and feature number of participants in a conversation.

(b) PSR* and feature emoticons.

(c) PSR* and feature correctly-spelled words.

(d) PSR* and feature sexual topic words.

(e) PSR* and feature time when a conversation starts.

(f) PSR* and feature intervention words per user.

(g) PSR* and feature sentiment and emotional markers.

Figure 24 – Evaluating the contribution of each of our proposed behavioral features. As it can be seen, our features contribute to increase accuracy in most percentages of information available. Note that features emoticons and correctly-spelled words contribute only when large portions of information are available; they become valuable when combined with our other five BFs since in this setting they do not harm accuracy if there is few information to be analyzed.

with feature sexual topic words, specifically when 10% of the information was available, but, as it can be seen in Figure 24d, this feature becomes a considerable asset as more information arrives.

Here, let us highlight the results obtained with feature sentiment and emotional markers. As it can be seen in Figure 24g, this feature is the one that produced more gain individually. For example, when only 10% of the information was available, the $\mathrm{F1}_g$ measure increased from 50% to 56.5%, and, when 90% of the information was available, the $\mathrm{F1}_g$ measure increased from 62.5% to nearly 70%.

To conclude this subsection, note that the experimental results confirm that all of our proposed behavioral features contribute to the early detection of online grooming. Throughout the analysis of the results, we observed that there are features that contribute more than the others, as in the case of feature sentiment and emotional markers. We also observed that there are features that begin to offer a considerable contribution when half or more of the total information is available. These results allow us to demonstrate that the analysis presented in Section **??** is correct. In the next subsection, we focus on Question **Q3** to investigate the gains obtained by taking advantage of all of our behavioral features combined.

7.3 Evaluating effectiveness with our proposed, final setting

This section answers Question **Q3** by demonstrating that our proposed framework BF-PSR outperforms PSR* and five of the best and recent previous approaches when evaluating the entire SGD corpus. From the previous subsections, we have concluded that: a) PSR* is the best approach in a scenario where behavioral features are disregarded, and; b) each of our features contributes to the detection of online grooming individually. Thus, we now analyze the contribution of all BFs combined with the profiles of PSR*.

Figure 23 reports the accuracy of our proposed framework BF-PSR and that of PSR*, ENB, PSR, MulR-W2V, MulR-PSR, and MulR-TVT. As it can be seen, our BF-PSR clearly outperforms all the other methods. Note that it also outperforms every one of the alternative methods studied in the previous subsection, i.e., those methods that take advantage of one behavioral feature at a time, whose results are shown in Figure 24. In this way, it is possible to confirm that the process of combining all the proposed BFs with the profiles of our modified PSR* allows us to achieve state-of-the-art results in the area of early detection of grooming. Furthermore, as it was informed in the previous subsection, features emoticons and correctly-spelled words become valuable in this setting since they did not harm accuracy with less information to be analyzed and still improved accuracy when there was more information available. We believe that it occurs because these two behavioral features offer a contrast between the classes that may be too small to be worth individually, but, they help to distinguish the classes when combined with the patterns present in the other features.

Continuing with a detailed analysis, we confirm once again that the previous methods are ineffective when the entire dataset is evaluated. It should be recalled that this is because the previous methods apply filtering techniques, so their performance decrease considerably when they must process highly unbalanced data that strongly resembles a real scenario where it is not possible to ignore a potential groomer conversation. On the other hand, our framework BF-PSR overcomes the data imbalance. When comparing BF-PSR with the previous methods, our proposal offers a gain of more than 40% $\mathrm{F1}_g$ with only 10% of the information available, which is highly desirable in a scenario of early classification; still, the gain is larger than 30% $\mathrm{F1}_g$ when 100% of the information is available. Additionally, when comparing our final proposal with PSR*, BF-PSR still offers a gain of approximately 10% $\mathrm{F1}_g$ through every single portion of text available. To the best of our knowledge, these results represent the state of the art in the area of early grooming detection.

7.4 Validating our BF-PSR framework with a different dataset than the SGD testing corpus

This section answers Question 4. As mentioned in the discussions of Subsection 4.2.4, it was decided to assemble two new datasets to mitigate the corpus shortage in the area of online grooming. Furthermore, we decided to validate our BF-PSR framework with such datasets and compare the obtained results with state-of-the-art methods. Following we describe in more detail how these datasets were assembled.

7.4.1 Assembling the PJZ dataset

Based on the distribution of the SGD dataset, we first had to collect groomer conversations to play the role of the TPs. For this purpose, we decided to extract conversations from the PJ website. However, it is important to emphasize that these conversations are completely disjoint from those in the SGD dataset, i.e., PJ chats $\not\subseteq$ SGD chats. Recall that the SGD corpus was published in 2012, which implied that we had to collect all the appropriate[2] conversations from the PJ website from the year 2013 onwards. Therefore, from the 622 groomers published on the PJ website, we collected the conversations of 24 of them. Then, we decided to collect the conversations that play the role of TNs, i.e., the non-groomer conversations. For this, we decided to focus on obtaining conversations with a topic similar to the TNs chats of the SGD corpus, which are computer-oriented conversations, e.g., HTML, CSS, apache, macOS, etc. To this end, we accessed a repository that hosts IRC logs for the #ZIG channel. These chats discuss the ZIG programming language and occur between two or more participants. Again, #ZIG chats $\not\subseteq$

[2] By appropriate we mean that the messages of a conversation must have the time of sending. Many of the conversations on the PJ website do not have this attribute, i.e., HH:MM.

SGD chats. The repository contains chats from 2017 and continues collecting chats; at the time of writing this paper, the repository contains chats up to October 2021. We will use the chats collected up to this date for the dataset assembly.

	PJZ dataset	PJZC dataset
PJ chats	1,104	1,104
ZIG chats	12,718	12,718
Chit chats	0	7,248
Total chats	13,822	21,070

Table 18 – Number of chats that compose the PJZ and PJZC datasets respectively.

After the data collection process, we iterated through each collected conversation and applied two modifications: a) If the sending time between the exchanged messages is longer than 25 minutes, this set of messages becomes a new conversation, and; b) If a conversation has more than 150 messages, it is removed from the set. These modifications were first applied by the authors of the SGD corpus. They observed that 25 minutes is a reasonable threshold for a topic change in the conversation and thus they generate a new one. Moreover, most of the conversations they collected had less than 150 messages. Because SGD is the most popular dataset in the online grooming area, we decided to follow these modifications to standardize our datasets. Table 18 describes that there are 1,104 PJ chats and 12,718 #ZIG chats when applying these criteria. Finally, we decided to gather all the chats, i.e., TPs + TNs, into a single dataset and refer to it as the PJZ dataset. Where PJ refers to the new conversations collected from the PJ website and Z refers to the conversations obtained from the #ZIG channel. The table also describes that there are a total of 13,822 conversations in the PJZ dataset. Finally, Figure 25a highlights the unbalance of the PJZ dataset where only 8% corresponds to groomer conversations; we aimed to keep this ratio because, as was explained, groomer conversations rarely occur in real life when compared to other types of conversations.

7.4.2 Assembling the PJZC dataset

For the assembly of this set, we wanted to address a broader range of topics in the conversations; a feature that is not well handled by the TNs of the SGD corpus because they are computer-oriented conversations. Thus, we decided to obtain the conversations from the Chit-Chat dataset which gathers thousands of conversations between peers of college students. It contains conversations with various topics that do not contain offensive behavior, e.g., What's one thing your best friend doesn't know about you?, Who is your modern-day hero?, Do you think computers will ever be given the right to vote?, etc. Following, as we did with the PJ and #ZIG chats, we removed all chats with more than 150 messages and generated new conversations whenever there was a break of more than 25 minutes. Thus, Table 18 describes that there are 7,248 Chit-Chat conversations. Finally, we added these conversations to the previously described dataset, the PJZ dataset, and created a new one which we named as PJZC, where the acronym C refers to the conversations of the Chit-Chat set. The table also describes that there are a total of 21,070 conversations in the PJZC dataset. Furthermore, Figure 25b highlights its unbalance. This time the percentage of the groomer conversations is 5.2%, turning it into an even more unbalanced dataset than the PJZ set.

7.4.3 Evaluating the performance of the BF-PSR framework whit the PJZ and PJZC datasets

After the assembly of both the PJZ and the PJZC datasets, we wanted to know how our proposal performs with data not included in the SGD testing set. To do so, the classifiers of the BF-PSR framework and the five best and recent state-of-the-art algorithms, i.e., ENB, PSR and the three variants of MulR, **continue to be trained with the SGD training set** and the new datasets, PJZ and PJZC, are used only for the testing phase; none of the new conversations were used for training. Then, as we did with the SGD testing set, we split both the PJZ and PJZC datasets into ten portions of text available to simulate the arrival of the incoming messages. Finally, we obtained the F1$_g$ measure with every portion of the information available, i.e. 10%, 20%,...,100% of the text available. Note that for the BF-PSR framework, we repeat the entire procedure described in Section 6.4 but now with PJZ and PJZC as testing sets.

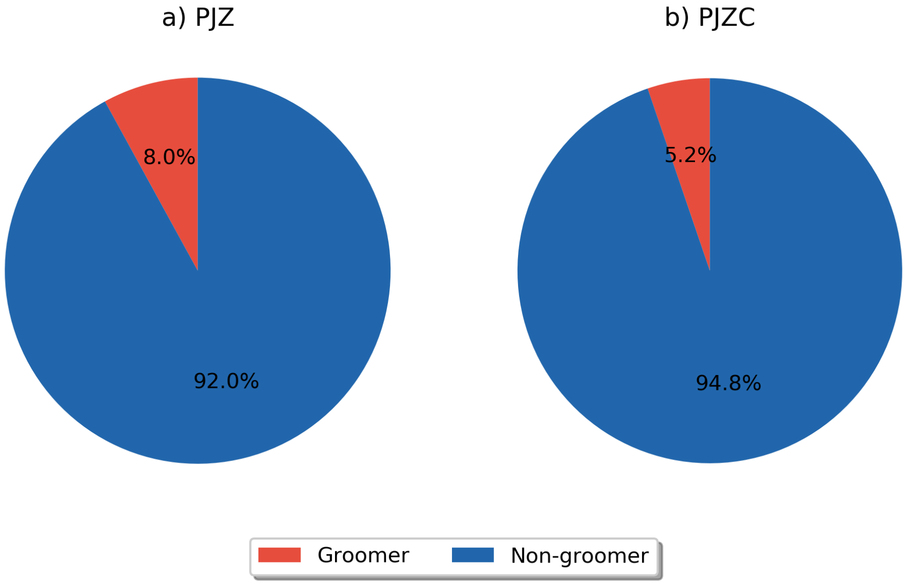


Figure 25 – Percentage of groomer and non-groomer conversations in the PJZ and PJZC datasets. In both sets, the G-conversations represent less than 10% of the whole corpus.

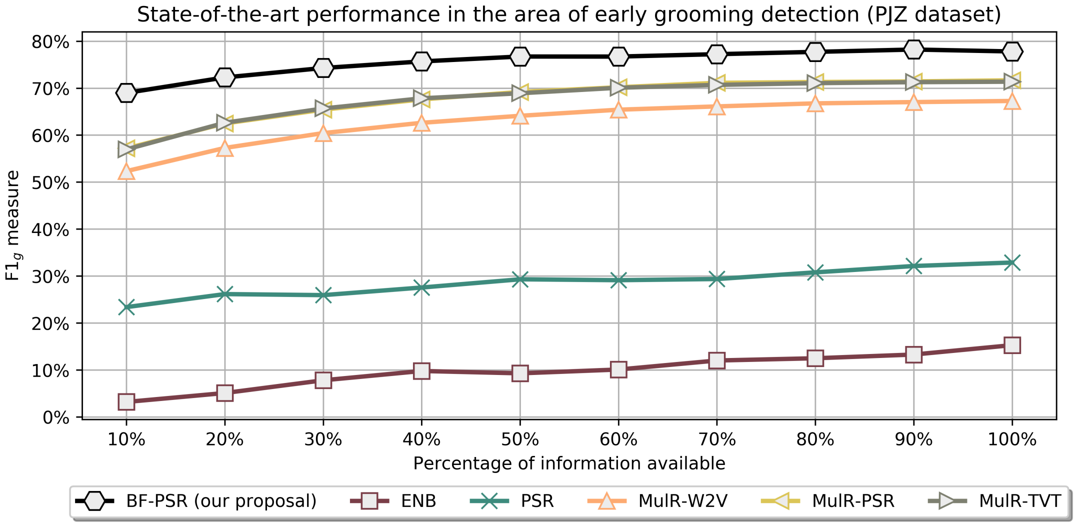
Figure 26a reports the accuracy of our proposed framework BF-PSR and that of the state-of-the-art methods. As it can be seen, ENB and PSR both struggle when processing a large amount of unbalanced data. ENB barely achieves 20% of F1$_g$ measure with 100% of the information available and PSR obtains just over 30% of F1$_g$ measure with all the available information. On the other hand, BF-PSR and the three variants of MulR provide much better results. This time, the MulR variants properly deal with the data imbalance. We believe it is because the PJZ dataset is easier to process. It contains conversations of opposite nature, i.e., groomer and computer-oriented conversations are easier to distinguish. However, we will note that this does not occur with a more complicated dataset such as the PJZC dataset. Thus, continuing with the analysis, MulR-W2V obtains about 50% of $F1_g$ measure with only 10% of the information available and methods MulR-PSR and MulR-TVT manage to obtain almost 60% of $F1_g$ measure with the same amount of text available. Nevertheless, our proposal clearly outperforms all the depicted methods. Even if the gain is not so evident as with the SGD testing set, our proposal achieves state-of-the-art results with the PJZ dataset obtaining about 70% of $F1_g$ measure with only 10% of the information available and improving its performance through every portion of text available.

Now, let us analyze the performance of all the previously described methods with the PJZC dataset. We can observe from Figure 26b that in this case, the results achieved by the algorithms are less accurate than those achieved with the SGD testing set and the PJZ dataset. This is because the TNs of both sets contain only computer-orientated conversations while the TNs of the PJZC dataset contains conversations with a wide variety of topics. This property causes this set to be much more complex to process and leads the classifiers trained **only** with the SGD training set to struggle during the classification. Thus, of all the methods depicted in the figure, ENB is again the worst performing method. It barely achieves 14% of $F1_g$ measure with 100% of the information available. On the other hand, the accuracy of PSR and the three variants of MulR, i.e., MulR-W2V, MulR-PSR, and MulR-TVT, fluctuates between 20% and 36% of $F1_g$ measure. Specifically, PSR obtains an approximate 20% of accuracy with 10% of the information available and nearly 30% with all the information available. While MulR-PSR, the best performing variant, obtains 26% of $F1_g$ measure with 10% of the information available and 36% $F1_g$ measure with all the information available. Finally and once more, BF-PSR outperforms the five methods from the state of the art. Specifically, our proposal obtains a gain in the $F1_g$ measure of about 15% through every single portion of the text available. This occurs despite processing a dataset of a more complex nature. Therefore, our proposal obtains state-of-the-art results when processing the PJZC dataset.

7.5 Summary and discussion

This chapter reported a series of experiments to evaluate the effectiveness of the most recent state-of-the-art preventive tools, and the impact of the modifications made to the PSR method to get our PSR* proposal. We also evaluated the contribution of each proposed BF individually and assessed the impact of concatenating the PSR* vector representation with the seven BFs. The results confirmed that our final method, the BF-PSR framework outperformed all the state-of-the-art methods, that is, the ENB, PSR, and the three MulR variants when processing the SGD corpus. Furthermore, we assembled two new datasets to mitigate the shortage of grooming data.

(a) PJZ is an easy-to-process dataset. The TPs are groomer conversations and the TNs are computer-oriented conversations. From the plot can be observed that methods ENB and PSR struggle when there is a large amount of disproportionate data. On the other hand, the three variants of MulR provide much better results, obtaining almost 60% of $F1_g$ measure with only 10% of the information available. Even so, our proposal manages to outperform all the methods in the state of the art. As it is expected, our behavioral features allow BF-PSR to obtain about 70% of $F1_g$ measure with only 10% of the information available and its performance improves through every portion of the information available. Therefore, our proposal BF-PSR obtains state-of-the-art performance with the PJZ dataset.



(b) PJZC is a more complex dataset. The TPs are groomer conversations and the TNs cover a wide range of topics. From the plot can be observed that ENB is the method with the worst performance reaching nearly 15% of $F1_g$ measure with all available information. The accuracy of PSR and the three MulR variants fluctuates between 20% and 36% of $F1_g$ measure. And once again, our proposal outperforms all the depicted methods. Our behavioral features allow BF-PSR to obtain a gain in the accuracy of about 15% through every single portion of the information available. Therefore, our proposal obtains state-of-the-art results when processing the PJZC dataset.

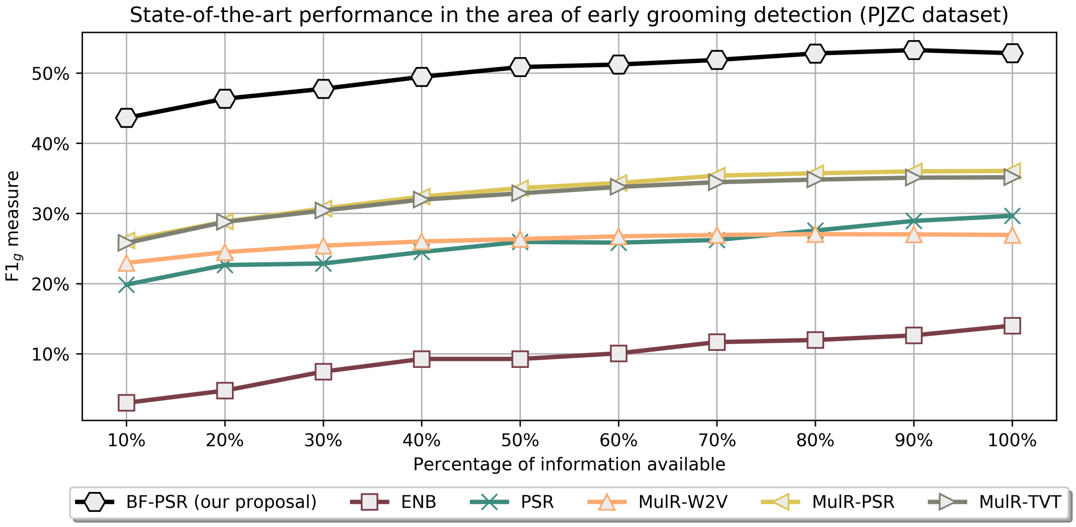


Figure 26 – Our BF-PSR framework and the five previous methods in the area of early detection of grooming are validated with the PJZ and PJZC datasets respectively. The subplots report the accuracy of results achieved by each method when different percentages of text are available for processing. Please note that all the methods were trained with the SGD training set only.

These are the PJZ and PJZC datasets. Then, with our last benchmark, we validated that once again BF-PSR outperformed all the described methods and obtained state-of-the-art results with such datasets. Note that these sets are a **contribution** to the area of online grooming detection. In turn, we are the first ones to validate a preventive scenario method with a dataset different from the SGD testing corpus. As a side note, please notice that we could not include conversations of sexual connotation between two adults in our datasets, i.e., the FPs. This is because we could not find such a collection of chats publicly available on the Internet. We hope to be able to collect this type of chats in the future. In the next section, we present the conclusions of our research and future work.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

This MSc work focused on the early detection of grooming in online conversations. Our contributions are:

C1 Behavioral features for the early detection of online grooming: A set of seven behavioral features was proposed to detect grooming behavior in online conversations. Specifically, our proposed features referred to the number of participants in a conversation, emoticons, correctly-spelled words, sexual topic words, the time when a conversation starts, intervention words per user, and sentiment and emotional markers. In addition, a detailed study was conducted to understand the reasons why each feature contributes to the task of early detection of grooming. The following is a summary of what was revealed in the study:

Number of participants in a conversation: By analyzing this feature, we discovered that there are three types of conversations in the SGD corpus, i.e., group, monologue, and pair. After analyzing each type of conversation in detail, we found that most of the monologue conversations of class groomer (G) represented fail attempts by groomer users to contact a victim. In addition, we found that groomers prefer to keep their victims isolated avoiding any kind of group conversation. As a result, monologue and pair conversations were the most relevant chats in the groomer class.

Sentiment and emotional markers: Through this feature was found an "avalanche" of emotions in the words employed by groomers. We believe that they use emotions such as joy or anticipation to create a trusting relationship with the victim and also to convince the child to meet offline and begin a sexual relationship.

Sexual topic words: After analyzing this feature, it was observed that proportionally, more sexually related terms are used in G-class conversations. These words are usually typos of the original word to avoid its detection by monitoring systems.

For example, for the sexual term “dick” a list of possible typos can be “dickk”, “dickkk” , “dickss”, etc.

Time when a conversation starts: The time when a conversation starts in the SGD corpus was analyzed, it was observed that from 4pm to 11pm there was a high flow in the *G*-class conversations. This indicates a potential risk in the children’s online interactions. We believe that this range of time reflects the children’s schedule after school, which explains their ease access to computers.

Correctly-spelled words: After analyzing the correctly-spelled words, we found out that they are more common in the groomer class conversations and generate enough bias compared to the non-groomer class to be used.

Emoticons: The most used emoticons were discovered in the groomer class, e.g., :) and :-*. Also, from the preprocessing step, we found out that removing the emoticons affects negatively the performance of the classifier. Therefore, maintaining the writing style of the authors such as the use of emoticons was of utmost importance.

Intervention words per user: For this feature, it was decided to weigh users’ participation in a conversation to know their interest in it. The results revealed that in *G*-class conversations there is a higher interest of users to keep the conversation going. This may reflect the groomers’ interest in maintaining the conversation with their victims.

Through this study, we shorten the gap existing between the forensic works and the works performing in a preventive scenario, since, to the best of our knowledge, we are the first ones to employ BFs for the early detection of grooming.

C2 **PSR*:** The PSR* method was proposed as a variant of the original PSR algorithm to continue representing the data in a non-sparse and low-dimensional vector space. However, the made adjustments allowed PSR* to obtain high-quality results despite working with a high number of unbalanced conversations. Some of these changes were: pass from three character gram to one token gram, choose TF.IDF as the official term-weighting schema and preserve the writing style of the author in the conversations, e.g., punctuation, emoticons, typos, etc. PSR* is considered one of our major contributions to this work.

C3 **BF-PSR framework:** The BF-PSR framework was introduced as an extension of the PSR framework in such a way that we adequately inserted our proposed behavioral features in a vector space, which remains non-sparse and low-dimensional as it is desirable, but now contains more valuable patterns that highlight the differences between groomer and non-groomer conversations. To achieve this, two main stages compose the pipeline of our proposal. **Offline stage:** Here, the training conversations were preprocessed and each one was represented by a PSR* vector. Then, the BFs were computed to highlight the behavioral patterns occurring in the different classes, i.e., groomer and non- groomer conversations, and to represent each conversation in a novel vector. Both vectors, PSR* and BFs, were then concatenated into a single vector named BF-PSR. Later, the classifier was trained with this new document representation space. **Online stage:** In this stage, the testing set was divided into ten portions containing different percentages of the information available, i.e., 10%, 20%, ..., and 100%. Then, the portions of information were preprocessed and represented by partial BF-PSR vectors. Finally, the classifier performed predictions with the early data provided.

C4 **Assembly of two new datasets:** To mitigate the shortage of data in the area of online grooming, we assembled and studied two new datasets, which we named as PJZ and PJZC.

PJZ: Gathers conversations from the Perverted Justice website as true positives and conversations from the IRC channel with the #ZIG query as true negatives. This dataset is considered easy to process because it has conversations of opposite nature, i.e., groomer conversations and conversations focused on the Zig programming language.

PJZC: This dataset attempts to address one of the shortcomings of the true negatives of the SGD dataset by including conversations with a wider variety of topics. Thus, the PJZC dataset adds day-to-day conversations from the Chit-Chat set to the TNs of the PJZ dataset. As a consequence, PJZC is more difficult to process by state-of-the-art methods.

C5 **Experimental evaluation:** To demonstrate that behavioral features truly contribute to the early detection of online grooming, we performed different experiments to evaluated the effectiveness of our proposal. As **first experiment**, we decided to evaluate the effectiveness of our PSR* proposal against the state-of-the-art methods of the preventive scenario, i.e., PSR and the three variants of method MulR. From the results, it was revealed that PSR* outperforms all methods using the complete SGD corpus. This indicates that the modifications performed to the original PSR were fundamental to overcome the data imbalance. In the **second experiment**, we evaluated the contribution of the BFs. To do this, we compared each behavioral feature individually and compared the results against the PSR* method. The results revealed that all the BFs contribute, even when there are features that contribute more than the others, as in the case of the sentiment and emotional markers BF. The **third experiment** evaluated the effectiveness of our final proposal. We compared the BF-PSR framework against the PSR*, PSR, MulR-W2V, MulR-PSR, and MulR-TVT methods. Through the results, it was revealed that our BF-PSR framework outperforms all the other methods. In this way, it was possible to confirm that the process of combining all the proposed BFs with the profiles of our modified PSR* allows us to achieve state-of-the-art results in the area of early detection of grooming.

8.1 Further discussions

- As an additional contribution, in Appendix A, we introduce the concept of a triggering mechanism and its appropriate metrics of evaluation. Such a mechanism decides "when" the information processed by the preventive system, e.g., the BF-PSR framework, is enough to perform a alert of potential grooming attack as early as possible. Additionally, a simple but effective triggering mechanism was proposed to complement our BF-PSR framework and evaluated it in a more realistic environment where the testing set was released message-by-message. After a series of experiments varying the parameters of the triggering mechanism, i.e., the minimum confidence value, promising results were obtained since the BF-PSR framework tends to perform alerts with few messages as is desired.
- Additionally to the triggering mechanism implementation, we verify if by applying text enrichment techniques we could increase the accuracy of the BF-PSR framework. Several techniques were tested such as replacing elongated words, (e.g., from `goood` to good), expanding contractions (e.g., from `don't` to do not), word segmentation (e.g., from `#goodvibes` to good vibes), etc. First, we decided to change all the preprocessing stated in Section , by one of the enrichment techniques, so if originally we only applied lowercase to the text, now we only applied the chosen technique to the whole corpus. This new enriched text was first used by both the PSR profiles and the BFs. However, the results obtained were worse than those achieved by the original BF-PSR. Then, as a second experiment, the enriched text was only used to extract the PSR profiles. Thus, the groomer and non-groomer profiles used the enriched text and the seven proposed BFs used the text with the original preprocessing. However, once again the results were not as good as expected. As a last experiment, in addition to the two PSR profiles and the seven proposed BFs under the original preprocessing, two new features were computed. These were obtained using the PSR method and the enriched text. Thus, we obtained a new groomer profile with enriched text and a new non-groomer profile with enriched text which were staked to the BF-PSR vector. However, despite the effort of generating various PSR profiles with the different techniques, the results were not promising. This leads us to conclude that the less preprocessing/enrichment techniques are applied to the SGD conversations, the more the author's writing style is preserved, which seems to be a determining factor for the detection of grooming behavior.

8.2 Future work

As future work, we aim to test our framework with datasets oriented to early solutions such as detection of bullying and scamming. We also want to implement indexing structures that improve the efficiency of the BF-PSR framework. Such structures can help the framework to be scalable and applicable in the real world where thousands of messages belonging to different conversations are sent at once. Additionally, we would like to confirm that behavioral features are also applicable to datasets with a different language such as Portuguese or Spanish. Finally, we consider the evolution of the written language as constant future work. Our framework as well as most of the related works in this monography employ datasets up to 10 years old, e.g., the SGD dataset, which does not guarantee that the systems are efficient with more recent online conversations. With the assembly of the PJZ and PJZC datasets, we demonstrate that our BF-PSR method continues to be accurate with recent language conversations, i.e., conversations collected since 2017. Despite this, language evolution remains an open problem to be taken into account when developing tools in the area of grooming detection.

To conclude, let us highlight that this MSc work generated the following publication:

- Daniela F. Milon-Flores, Robson L.F. Cordeiro - How to take advantage of behavioral features for the early detection of grooming in online conversations. In: Knowledge-Based Systems, Elsevier. Volume 240, 2022, 108017, ISSN: 0950-7051. DOI: <https://doi.org/10.1016/j.knosys.2021.108017>.

BIBLIOGRAPHY

BERNARDINI, G.; CAMILLI, S.; QUAGLIARINI, E.; D'ORAZIO, M. Flooding risk in existing urban environment: from human behavioral patterns to a microscopic simulation model. **Energy Procedia**, v. 134, p. 131 – 140, 2017. ISSN 1876-6102. Sustainability in Energy and Buildings 2017: Proceedings of the Ninth KES International Conference, Chania, Greece, 5-7 July 2017. Available: <http://www.sciencedirect.com/science/article/pii/S1876610217346805>. Citation on page 58.

BOGDANOVA, D.; ROSSO, P.; SOLORIO, T. Exploring high-level features for detecting cyberpedophilia. **Comput. Speech Lang.**, Academic Press Ltd., GBR, v. 28, n. 1, p. 108–120, Jan. 2014. ISSN 0885-2308. Available: <https://doi.org/10.1016/j.csl.2013.04.007>. Citations on pages 50 and 59.

BOURS, P.; KULSRUD, H. Detection of cyber grooming in online conversation. In: **2019 IEEE International Workshop on Information Forensics and Security (WIFS)**. [S.l.: s.n.], 2019. Citations on pages 13 and 45.

BURDISSO, S. G.; ERRECALDE, M.; MONTES-Y-GÓMEZ, M. A text classification framework for simple and effective early depression detection over social media streams. **Expert Syst. Appl.**, v. 133, p. 182–197, 2019. Available: <https://doi.org/10.1016/j.eswa.2019.05.023>. Citation on page 52.

CARDEI, C.; REBEDEA, T. Detecting sexual predators in chats using behavioral features and imbalanced learning. **Natural Language Engineering**, Cambridge University Press, v. 23, n. 4, p. 589–616, 2017. Citations on pages 50 and 59.

Cheong, Y.; Jensen, A. K.; Guðnadóttir, E. R.; Bae, B.; Togelius, J. Detecting predatory behavior in game chats. **IEEE Transactions on Computational Intelligence and AI in Games**, v. 7, n. 3, p. 220–232, 2015. Citations on pages 50 and 59.

CRIJNS, T. **Classifying events to ugenda calendar genres**. 31 p. Master's Thesis (Master's Thesis) — Radboud University, 2016. Citations on pages 29 and 37.

DUCHI, J.; HAZAN, E.; SINGER, Y. Adaptive subgradient methods for online learning and stochastic optimization. **J. Mach. Learn. Res.**, JMLR.org, v. 12, n. null, p. 2121–2159, Jul. 2011. ISSN 1532-4435. Citation on page 40.

EBRAHIMI, M.; SUEN, C. Y.; ORMANDJIEVA, O. Detecting predatory conversations in social media by deep convolutional neural networks. **Digital Investigation**, v. 18, p. 33 – 49, 2016. ISSN 1742-2876. Available: <http://www.sciencedirect.com/science/article/pii/S1742287616300731>. Citations on pages 50 and 59.

EDWARDS, A.; LEATHERMAN, A. Chatcoder: Toward the tracking and categorization of internet predators. v. 3, 01 2009. Citations on pages 48 and 59.

ERRECALDE, M. L.; VILLEGAS, M. P.; FUNEZ, D. G.; UCELAY, M. J. G.; CAGNINA, L. C. Temporal variation of terms as concept space for early risk prediction. In: CAPPELLATO, L.;

FERRO, N.; GOEURIOT, L.; MANDL, T. (Ed.). **Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017**. CEUR-WS.org, 2017. (CEUR Workshop Proceedings, v. 1866). Available: <http://ceur-ws.org/Vol-1866/paper_103.pdf>. Citation on page 56.

ESCALANTE, H. J.; GOMEZ, M. Montes y; VILLASENOR, L.; ERRECALDE, M. L. Early text classification: a naïve solution. In: **Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis**. San Diego, California: Association for Computational Linguistics, 2016. p. 91–99. Available: <https://www.aclweb.org/anthology/W16-0416>. Citations on pages 28, 53, 55, 56, and 59.

ESCALANTE, H. J.; VILLATORO-TELLO, E.; GARZA, S. E.; LPEZ-MONROY, A. P.; GMEZ, M. Montes-y; VILLASEOR-PINEDA, L. Early detection of deception and aggressiveness using profile-based representations. **Expert Syst. Appl.**, Pergamon Press, Inc., USA, v. 89, n. C, p. 99–111, Dec. 2017. ISSN 0957-4174. Available: <https://doi.org/10.1016/j.eswa.2017.07.040>. Citations on pages 13, 26, 27, 28, 54, 55, 56, and 59.

Fauzi, M. A.; Bours, P. Ensemble method for sexual predators identification in online chats. In: **2020 8th International Workshop on Biometrics and Forensics (IWBF)**. [S.l.: s.n.], 2020. p. 1–6. Citations on pages 51 and 59.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citation on page 39.

GRON, A. **Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. 1st. ed. [S.l.]: O'Reilly Media, Inc., 2017. ISBN 1491962291. Citations on pages 13, 39, and 41.

HINTON, G. **Lecture 6e rmsprop: Divide the gradient by a running average of its recent magnitude**. 2020. Accessed: 2020–11-06. Citation on page 40.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural Comput.**, MIT Press, Cambridge, MA, USA, v. 9, n. 8, p. 1735–1780, Nov. 1997. ISSN 0899-7667. Available: <https://doi.org/10.1162/neco.1997.9.8.1735>. Citation on page 40.

INCHES, G.; CRESTANI, F. Overview of the international sexual predator identification competition at pan-2012. In: . [S.l.: s.n.], 2012. Citation on page 43.

JAISWAL, S. **Machine Learning Random Forest Algorithm**. 2021. Available: <https://www.javatpoint.com/machine-learning-random-forest-algorithm>. Citations on pages 13 and 38.

KADAM, S. H. **Text analysis for email multi label classification**. 37 p. Master's Thesis (Master's Thesis) — Gothenburg University, 2020. Citations on pages 29, 33, 34, and 36.

KETTUNEN, K.; KUNTTU, T.; JäRVELIN, K. To stem or lemmatize a highly inflectional language in a probabilistic ir environment? **Journal of Documentation**, v. 61, p. 476–496, 08 2005. Citation on page 31.

KIEFER, J.; WOLFOWITZ, J. Stochastic Estimation of the Maximum of a Regression Function. **The Annals of Mathematical Statistics**, Institute of Mathematical Statistics, v. 23, n. 3, p. 462 – 466, 1952. Available: <https://doi.org/10.1214/aoms/1177729392>. Citation on page 40.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. In: BENGIO, Y.; LECUN, Y. (Ed.). **3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings**. [s.n.], 2015. Available: <http://arxiv.org/abs/1412.6980>. Citation on page 40.

KONTOSTATHIS, A.; GARRON, A.; REYNOLDS, K.; WEST, W.; EDWARDS, L. Identifying predators using chatcoder 2.0. In: **CLEF**. [S.l.: s.n.], 2012. Citations on pages 49 and 59.

la Fuente Garcia, S. d.; Haider, F.; Luz, S. Cross-corpus feature learning between spontaneous monologue and dialogue for automatic classification of alzheimer's dementia speech. In: **2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)**. [S.l.: s.n.], 2020. p. 5851–5855. Citation on page 58.

LÓPEZ-MONROY, A. P.; GONZÁLEZ, F. A.; MONTES, M.; ESCALANTE, H. J.; SOLORIO, T. Early text classification using multi-resolution concept representations. In: **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)**. New Orleans, Louisiana: Association for Computational Linguistics, 2018. p. 1216–1225. Available: <https://www.aclweb.org/anthology/N18-1110>. Citations on pages 13, 28, 52, 55, 57, and 59.

LORENZO-DUS, N.; KINZEL, A. **Pre publication draft - 'So is your mom as cute as you?'- Examining patterns of language use in online sexual grooming of children**. 2019. Citation on page 47.

LOSADA, D.; CRESTANI, F. A test collection for research on depression and language use. In: **Proc. of Experimental IR Meets Multilinguality, Multimodality, and Interaction, 7th International Conference of the CLEF Association, CLEF 2016**. Evora, Portugal: [s.n.], 2016. p. 28–39. Citations on pages 14, 112, and 113.

LOSADA, D. E.; CRESTANI, F.; PARAPAR, J. Overview of erisk at CLEF 2020: Early risk prediction on the internet (extended overview). In: CAPPELLATO CARSTEN EICKHOFF, N. F. L. (Ed.). **Conference and Labs of the Evaluation Forum**. [S.l.]: CEUR Workshop Proceedings, 2020. ISSN 1613-0013. Citations on pages 115 and 116.

LYKOUSAS, N.; PATSAKIS, C. **Large-scale analysis of grooming in modern social networks**. 2020. Citations on pages 51 and 59.

MARTIN, S.; NELSON, B.; SEWANI, A.; CHEN, K.; JOSEPH, A. Analyzing behavioral features for email classification. In: . [S.l.: s.n.], 2005. Citation on page 58.

MCGHEE, I.; BAYZICK, J.; KONTOSTATHIS, A.; EDWARDS, L.; MCBRIDE, A.; JAKUBOWSKI, E. Learning to identify internet sexual predation. **International Journal of Electronic Commerce**, Routledge, v. 15, n. 3, p. 103–122, 2011. Available: <https://doi.org/10.2753/JEC1086-4415150305>. Citations on pages 48, 50, and 59.

MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G.; DEAN, J. Distributed representations of words and phrases and their compositionality. In: **Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2**. Red Hook, NY, USA: Curran Associates Inc., 2013. (NIPS'13), p. 3111–3119. Citation on page 56.

MORRIS, C.; HIRST, G. Identifying sexual predators by svm classification with lexical and behavioral features. In: **Working notes of the CLEF 2012 Evaluation Labs and Workshop**. [S.l.: s.n.], 2012. Citations on pages 26, 48, 49, and 59.

MUCHERINO, A.; PAPAJORGJI, P. J.; PARDALOS, P. M. k-nearest neighbor classification. In: ____. **Data Mining in Agriculture**. New York, NY: Springer New York, 2009. p. 83–106. ISBN 978-0-387-88615-2. Available: <https://doi.org/10.1007/978-0-387-88615-2_4>. Citation on page 40.

MYERS, W.; ETCHART, T.; FULDA, N. Conversational scaffolding: An analogy-based approach to response prioritization in open-domain dialogs. 2020. Citation on page 60.

NASEEM, U.; RAZZAK, I.; EKLUND, P. W. A survey of pre-processing techniques to improve short-text quality: a case study on hate speech detection on twitter. **Multimedia Tools and Applications**, p. 1–28, 2020. Citations on pages 17, 29, 30, 31, and 32.

PARAPAR, J.; LOSADA, D.; BARREIRO, A. A Learning-based Approach for the Identification of Sexual Predators in Chat Logs—Notebook for PAN at CLEF 2012. In: FORNER, P.; KARLGREN, J.; WOMSER-HACKER, C. (Ed.). **CLEF 2012 Evaluation Labs and Workshop – Working Notes Papers, 17-20 September, Rome, Italy**. CEUR-WS.org, 2012. ISBN 978-88-904810-3-1. ISSN 2038-4963. Available: <http://www.clef-initiative.eu/publication/working-notes>. Citations on pages 49 and 59.

PENDAR, N. Toward spotting the pedophile telling victim from predator in text chats. In: **Proceedings of the International Conference on Semantic Computing**. USA: IEEE Computer Society, 2007. (ICSC '07), p. 235–241. ISBN 0769529976. Available: <https://doi.org/10.1109/ICSC.2007.102>. Citations on pages 48 and 59.

QUINLAN, J. R. Induction of decision trees. **Mach. Learn.**, Kluwer Academic Publishers, USA, v. 1, n. 1, p. 81–106, Mar. 1986. ISSN 0885-6125. Available: <https://doi.org/10.1023/A:1022643204877>. Citation on page 40.

RANGEL, F.; ROSSO, P.; KOPPEL, M.; STAMATATOS, E.; INCHES, G. Overview of the Author Profiling Task at PAN 2013. In: FORNER, P.; NAVIGLI, R.; TUFIS, D. (Ed.). **CLEF 2013 Evaluation Labs and Workshop – Working Notes Papers, 23-26 September, Valencia, Spain**. CEUR-WS.org, 2013. ISBN 978-88-904810-3-1. ISSN 2038-4963. Available: <http://ceur-ws.org/Vol-1179>. Citation on page 51.

ROA, D. **Analysis of Short Text Classification strategies using Out-of-domain Vocabularies**. 53 p. Master's Thesis (Master's Thesis) — KTH Royal Institute of Technology, 2018. Citation on page 34.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning Representations by Back-propagating Errors. **Nature**, v. 323, n. 6088, p. 533–536, 1986. Available: <http://www.nature.com/articles/323533a0>. Citation on page 39.

SADEQUE, F.; XU, D.; BETHARD, S. Measuring the latency of depression detection in social media. In: **Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining**. New York, NY, USA: Association for Computing Machinery, 2018. (WSDM '18), p. 495–503. ISBN 9781450355810. Available: <https://doi.org/10.1145/3159652.3159725>. Citations on pages 14, 113, 115, and 117.

SARKAR, D. **Text Analytics with Python: A Practitioner's Guide to Natural Language Processing**. 2nd. ed. [S.l.]: APress, 2019. ISBN 1484243536. Citations on pages 13, 29, 36, 37, 38, 40, and 41.

TROTZEK, M.; KOITKA, S.; FRIEDRICH, C. M. Utilizing neural networks and linguistic metadata for early detection of depression indications in text sequences. **CoRR**, abs/1804.07000, 2018. Available: <http://arxiv.org/abs/1804.07000>. Citations on pages 113 and 114.

VILLATORO-TELLO, E.; JUÁREZ-GONZÁLEZ, A.; ESCALANTE, H.; GÓMEZ, M. M. y; VILLASEÑOR-PINEDA, L. A two-step approach for effective detection of misbehaving users in chats notebook for pan at clef 2012. In: . [S.l.: s.n.], 2012. Citations on pages 45, 49, 51, and 59.

ZUO, Z.; LI, J.; ANDERSON, P.; YANG, L. Grooming detection using fuzzy-rough feature selection and text classification. In: **2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)**. [S.l.]: IEEE, 2018. p. 1–8. 2018 IEEE International Conference on Fuzzy Systems, FUZZ- IEEE 2018 ; Conference date: 08-07-2018 Through 13-07-2018. Citations on pages 51 and 59.

APPENDIX

A

TRIGGERING MECHANISM IN THE AREA OF EARLY DETECTION OF GROOMING

A.1 Introduction

In this monograph, the BF-PSR framework oriented to the early detection of grooming was proposed. This framework focuses on detecting grooming behavior in online conversations as early as possible, and without having a significant loss in terms of accuracy. However, to assess its efficiency, the testing dataset was divided into portions of text available to simulate the sequential arrival of messages. Partitioning the data assumes that we know the total number of messages of a conversation in advance, which is not the case in real applications. Thus, to employ the BF-PSR framework in a more realistic environment, we can employ a triggering mechanism and analyze each arriving message of the conversation as occurs in real life.

A triggering mechanism decides "when" the classifier outputs a reliable prediction with as little text as possible. This means that the preventive system, i.e., the BF-PSR framework, is ready to perform an alert and stop reading the conversation's incoming messages. To evaluate this type of mechanism, new metrics such as ERDE or $F_{latency}$ are proposed in the literature to penalize the late detection of positive cases in the testing set, i.e., the groomer conversations. Also, we found health and safety task-oriented related works, e.g., systems focused on the early detection of depression, anorexia, suicide, etc., that proposed different types of triggering mechanisms, however, none of them are focused on the early detection of grooming. Because of that, we decided not to describe these works; and concentrate our efforts on proposing a triggering mechanism in our area of concern, since as far as we know, we are the first ones to employ such a mechanism in the grooming detection area.

In the following sections, we first describe the appropriate metrics to penalize the late detection of true positives. Then, we present our proposed triggering mechanism oriented to the early detection of grooming. And finally, we employ such mechanism in conjunction with the BF-PSR framework to evaluate its efficiency in a more realistic environment.

A.2 Metric evaluations

Traditional metrics such as Accuracy, Recall, and $F1_g$, described in Chapter 2, serve to evaluate the output of the preventive system against the true labels in the dataset. However, these metrics are not focused on evaluating the response time, i.e., they are time-unaware, which is an important characteristic when employing a triggering mechanism. Because of that, next, we describe the evaluation metrics proposed so far in the literature to address this demand.

A.2.1 Early risk detection error

The Early Risk Detection Error (ERDE) measure was proposed by Losada and Crestani (2016) to evaluate the CLEF 2017[1] eRisk pilot task focused on the *early risk detection of depression*. The dataset used in this task is a collection of posts published by a user in chronological order. However, because we employ the SGD corpus, in this work we decided to adapt the ERDE metric to evaluate the messages of a conversation instead of the user's posts as in the original task. With that in mind, ERDE takes into account the correctness of the (binary) decision and the delay taken by the system to make the decision. Such delay is measured by counting the number (k) of distinct textual items seen before answering, e.g., the number of messages contained in a portion of text available. The ERDE metric is shown in Equation A.1.

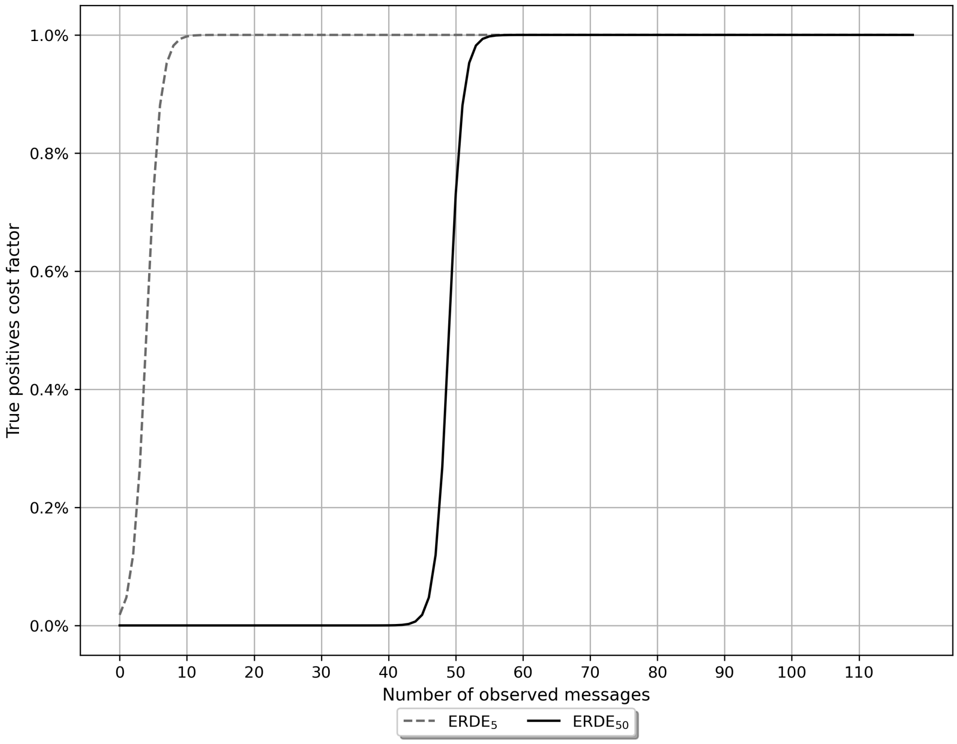
$$ERDE_o(d,k) = \begin{cases} c_{fp} & \text{if } d = \text{positive \& label} = \text{negative (FP)} \\ c_{fn} & \text{if } d = \text{negative \& label} = \text{positive (FN)} \\ lc_o(k) \cdot c_{tp} & \text{if } d = \text{positive \& label} = \text{positive (TP)} \\ 0 & \text{if } d = \text{negative \& label} = \text{negative (TN)} \end{cases} \tag{A.1}$$

The method takes as input the decision d taken by the system, i.e., the triggering mechanism in conjunction with BF-PSR, at point k and returns a penalty cost. The penalty is c_{fp}, if the prediction is a FP, i.e., the prediction is positive, but the label is negative. The penalty is c_{fn}, if the prediction is an FN, i.e., the prediction is negative, but the label is positive. Because frequently the positive class is much lower than the negative class, the value of c_{fn} is set to 1 in the paper, and the value of c_{fp} is set to the proportion of positive cases in the corpus, i.e., $\frac{n_G}{n}$, where n_G is the number of groomer conversations in the corpus and n is the number of total conversations. This setting aims to avoid building trivial classifiers that always say "no". If the classifier output is a TP, i.e., the prediction is positive and the label is positive, the method penalizes with a specific cost according to the delay of detecting positive cases, i.e., writings with potential signals of risk in health and safety. Equation A.2 describes this cost.

$$lc_o(k) = 1 - \frac{1}{1 + e^{k-o}} \tag{A.2}$$

[1] https://erisk.irlab.org/2017/index.html

Figure 27 – Plot of the true positive cost factor $lc_o(k)$ for $ERDE_5$ and $ERDE_{50}$. Based on (LOSADA; CRESTANI, 2016). The plot reveals that with o=5 the penalization is more severe than with o=50.



The latency cost function, i.e., $lc_o(k)$, is designed to grow with k, so; the longer the system takes to detect a TP, the higher the cost of the penalty. The parameter o controls around which point the (sigmoid) cost function is centered. The proposed values for o in the paper are 5 and 50. Figure 27 illustrates the different penalization of the TPs when parameter o varies. It can be seen that with $ERDE_5$ the penalty is more severe than with $ERDE_{50}$. Then, this value is multiplied by $c_{tp} = c_{fn} = 1$, i.e., a late detection is equivalent to not detecting the case at all. Finally, if the output is a TN, i.e., the prediction is negative and the label is negative, there is no penalty. The ERDE output is a value between $[0,1]$. Since there are n conversations in the corpus, the overall error would be the mean of the n ERDE values.

A.2.1.1 Discussions

Despite being one of the first metrics proposed to properly evaluate a triggering system, some problems have been found throughout the various research papers employing it. In the works of Trotzek, Koitka and Friedrich (2018), Sadeque, Xu and Bethard (2018), it is revealed that the penalty associated with true positives grows quickly to 1, being the transition between no penalty and 100% penalty very extreme. This is due to the functional form of the (sigmoid) cost function. Also in the work of Sadeque, Xu and Bethard (2018), it is proved that a perfect system, that is, a system that detects all TPs with $k = 1$ does not get an error equal to 0, e.g., $lc_o(k) = 1 - \frac{1}{1+e^{1-5}} > 0$. Other drawbacks of the metric are pointed out, such as the large number of parameters that must be set manually, e.g., o,c_{tp},c_{fn},c_{fn}, and that, ERDE values are not easily interpretable. In the following subsections, we describe other time-aware metrics that aim to tackle these disadvantages.

A.2.2 Percentage early risk detection error

The ERDE metric was first used with partitioning data that simulates the sequential arrival of information, i.e., the testing set was divided into ten portions of text available, where the first portion contains the oldest 10% of the information, the second portion contains the second oldest 10%, and so on. Nevertheless, when the penalty is performed with the ERDE metric, it takes into account the number of k messages in each portion to perform the alert. So, while for conversation c_x the first portion may contain three messages, for conversation c_y the first portion may contain a hundred or more messages and the penalization given to the conversations would not be appropriate. An example of this problem is illustrated in Figure 27. One can observe that positive cases with about ten and more messages per portion in $ERDE_5$, basically cannot be evaluated correctly because the cost would be very close to 1. The same occurs when there are about 55 and more messages per portion in $ERDE_{50}$. Thus, the authors of Trotzek, Koitka and Friedrich (2018), decided to take into account the **percentage** of the content of each portion before given the alert so that the penalty can be normalized. To do so, they proposed a modification of the original ERDE metric called $ERDE_o^{\%}$ by updating the latency cost function. Such modification is described in Equation A.3 and Equation A.4.

$$p = \frac{100 \cdot k}{n_m} \qquad \text{(A.3)} \qquad\qquad lc_o(p) = 1 - \frac{1}{1 + e^{p-o}} \qquad \text{(A.4)}$$

As it can be seen, the value of p (Equation A.3) represents the percentage of the messages read up to point k concerning the total number of messages (n_m) in a given conversation. Consequently, the penalization is a more intuitive cost that grows equally for every conversation independent of the number of messages. Additionally, the cost can still be parameterized by o (Equation A.4), and since $o = 18$ is the minimum value to achieve an error of 0.00% in the proposed function, the authors suggest using $ERDE_{20}^{\%}$ and $ERDE_{50}^{\%}$ for the evaluation of the triggering mechanism.

A.2.2.1 Discussions

$ERDE_o^{\%}$ aims to face the problem of a quickly growing penalty by normalizing the all the messages contributions. However, the metric was adapted to be used only with data in portions. This is a huge limitation if we want to evaluate real-life applications since it is not possible to know in advance the total number of messages in a conversation. In the following, we describe a metric that tackles this problem in real-life applications.

A.2.3 Latency and latency-weighted F1

Latency and latency-weighted F1 metrics, proposed in (SADEQUE; XU; BETHARD, 2018), aim to address the various drawbacks found in the ERDE and $ERDE_o^{\%}$ metrics by smoothing the growth of penalties, recognizing a perfect system by assigning error zero to it, normalizing the contributions in each conversation, and obtaining interpretable results. Besides, these metrics are not limited to use data in portions. They are designed to evaluate information item by item, i.e., evaluate message by message of a conversation, so it is not necessary to know the total number of messages in a conversation right away.

Latency Let us first describe the latency metric which aims to answer the following question: *how many messages should the preventive system observe before predicting a groomer conversation?*

$$latency_{TP} = \text{median}\{k : c \in C; d = g = 1\} \tag{A.5}$$

Formally, latency metric[2] is described in Equation A.5. It considers each conversation c in a set of conversations C to analyze if the prediction performed with k messages, where $k >= 1$, is a true positive case, i.e., the label $g = 1$ and the prediction $d = 1$. Then, the metric calculates the median of all the TPs found by the preventive system, e.g., BF-PSR framework in collaboration with the triggering mechanism, answering how many messages should the system expect to perform an alert. Consequently, the results are now interpretable. Note that unlike the original equation presented in Sadeque, Xu and Bethard (2018), in the above equation, latency$_{TP}$ only takes into account the TPs. This modification was proposed by the authors of Losada, Crestani and Parapar (2020) since the preventive system should not generate an alert if the prediction is a FN, i.e., the label $g = 1$ and the prediction $d = 0$, as is done in the original metric.

Latency-weighted F1 To give a complete picture of a system's performance, the authors propose the latency-weighted F1 measure also called F$_{latency}$ to produce a single metric that combines the F1$_g$ measure with a set of values that penalize the delay of positive cases detection. Equation A.6 describes this penalization.

$$penalty(k) = -1 + \frac{2}{1 + e^{-p\cdot(k-1)}} \tag{A.6}$$

The equation is applied for each TP case detected by the system and receives as input the number of k messages used to perform the decision. Also, similar to the o parameter in ERDE, p is a parameter that determines how quickly the penalty increases. In the original paper, p was set such that the penalty equals 0.5 at the median number of writings of a user. So, to calculate p in the SGD corpus we need to calculate the median of the TPs in the training set. To do this, we iterate each conversation $c \in C$ belonging to the groomer class, i.e., $g = 1$, and count the total number of messages n_m in each conversation. Then the median of these values is calculated

[2] Recall that to adapt the metrics to the SGD corpus, some terminology has been changed from the original metric description such as “post and user ” to “message and conversation”.

yielding in 5. Equation A.7 defines this procedure.

$$median_{TP} = \text{median}\{n_m : c \in C; g = 1\} \tag{A.7}$$

Later, we replace this value in the formula A.6 as $k = median_{TP} = 5$ to obtain the value of p which is 0.27, i.e., $0.5 = -1 + \frac{2}{1+e^{-p\cdot(5-1)}}$. To verify that the value of p is correct, Figure 28 illustrates that with 5 observed messages in a groomer conversation the penalty is 0.5. Thus, if the system needs more than 5 messages to predict a groomer conversation, the penalty approaches 1. On the other hand, if the system needs less than 5 messages, the penalty approaches 0. In this way, the proposed cost function aims to smooth the growth of the penalization. Following, the preventive system's speed factor is computed in Equation A.8:

$$speed = (1 - median\,\{penalty(k) : c \in C; d = g = 1\}) \tag{A.8}$$

The *speed* score calculates the median of the penalties of all the TPs detected by the system. The *speed* value equals 1 for a system whose TPs are detected right at the first message, i.e., $k = 1$. A slow system, which detects true positives after various messages, will be assigned a *speed* near 0 (LOSADA; CRESTANI; PARAPAR, 2020). Finally, the latency-weighted F1 score is depicted in Equation A.9.

$$F_{latency} = \text{F1}_g \text{ measure} \cdot speed \tag{A.9}$$

The $F_{latency}$ metric is the result of multiplying the F1_g measure and the system *speed*. Recall that the objective of $F_{latency}$ is to combine the efficiency of the preventive system, i.e., F1_g measure, with the penalty of late detection of TPs, i.e., *speed*, to give a complete picture of the system performance.

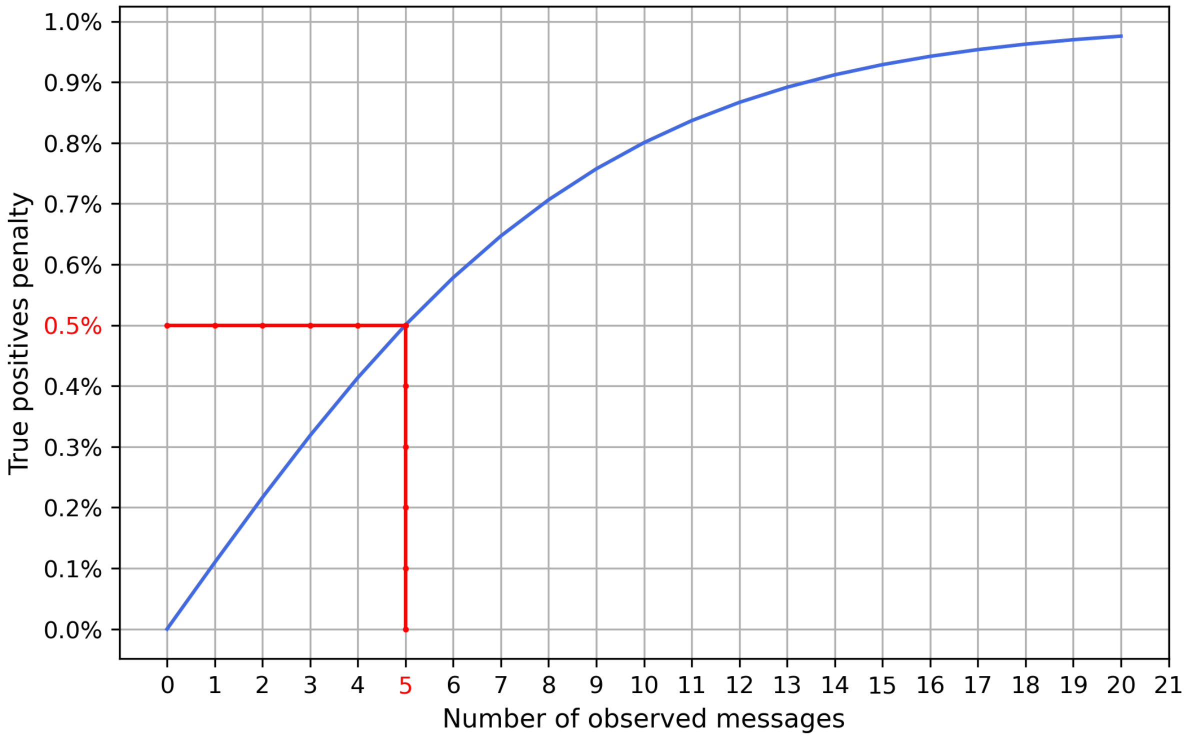
A.2.3.1 Discussions

Because $latency_{TP}$ and $\text{F}_{latency}$ succeed in addressing the limitations of the ERDE and $ERDE_o^{\%}$ metrics, in this monograph we decided to employ them to evaluate our triggering mechanism. Besides, as mentioned before, to adapt the metrics to the SGD corpus, some values were modified, such as the value of p in the *penalty* function. In the next section, we present our triggering mechanism to evaluate our preventive system, the BF-PSR framework, in a more realistic environment.

A.3 Proposed triggering mechanism

In this section, we propose a simple but effective triggering mechanism to complement the evaluation of the BF-PSR framework in a more realistic environment. Figure 29 illustrates the pipeline of how to employ this mechanism in combination with the BF-PSR framework. As a first observation, we move away from partitioning data in portions to message-by-message realising data. Thus, the BF-PSR framework waits for new incoming information, i.e., the messages of a

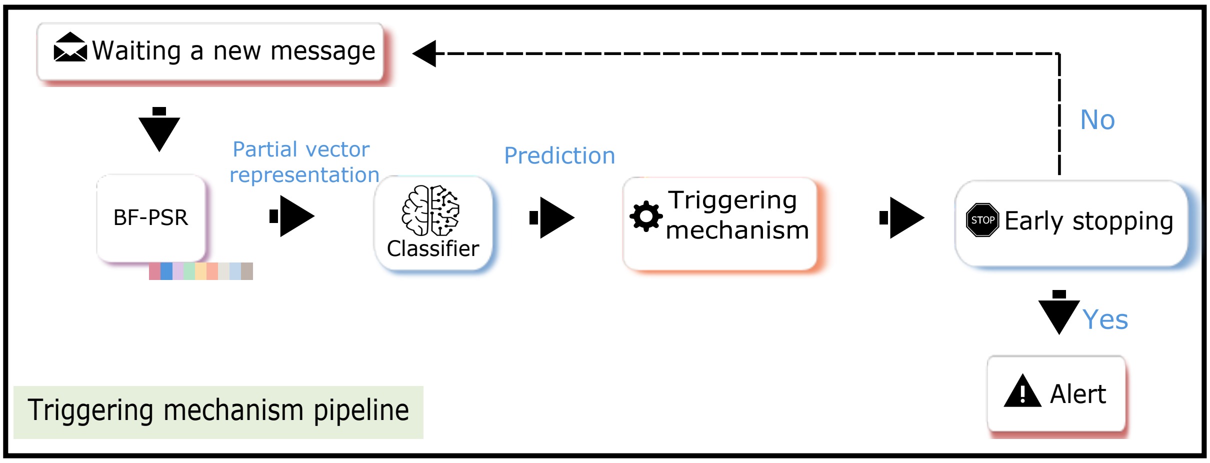
Figure 28 – Plot of how the penalty increases with the number of observed messages by the system. Based on (SADEQUE; XU; BETHARD, 2018)



conversation. After a new message arrives, the BF-PSR vector representation, i.e., the groomer and non-groomer profiles and the seven proposed BFs, is calculated. The vector is then passed as input to the classifier which predicts whether the messages read so far belong to a groomer conversation or not. Later, the triggering mechanism analyzes if the performed predictions are reliable enough and output a "early stopping" value. If this value is "positive", the incoming messages are not analyzed and an alert is immediately produced notifying a possible grooming attack. On the other hand, if the early stopping value is "negative", we proceed to wait for more information. Remember that the more information is required to perform an alert, the higher is the penalty for the preventive system.

We now proceed to explain how our triggering mechanism works in more detail. For this, we refer to Algorithm 3 which describes it formally. The triggering mechanism expects as input the predictions performed by the BF-PSR framework, i.e., $new\_p_{(i)}$, $previous\_p_{(i-1)}$, and the number of messages observed so far, i.e., $delay$. Besides, it has a single handset parameter, the $min\_confidence$ value which we will set manually in the following section to find its ideal value. This parameter represents how confident the mechanism is about the predictions performed by the preventive system. If $min\_confidence = 1$, no analysis is performed and the mechanism returns the BF-PSR predictions with only **one** message as delay (Line 2-4). If instead, $min\_confidence > 1$, the local $confidence$ value must increase to satisfy the minimum condition. There are two ways to make the local $confidence$ value increase. First, if the new incoming

Figure 29 – Pipeline of our proposed triggering mechanism. The BF-PSR framework waits for incoming information. When a new message arrives, it outputs the partial vector representation and the classifier predicts whether the messages read so far belong to a groomer or non-groomer conversation. The predictions are analyzed by the triggering mechanism which decides if more information is needed, i.e., early stopping = "No" or if the read information is sufficient to trigger an alert, i.e., early stopping = "Yes".



prediction increases concerning the previous prediction, i.e., $new\_p_{(i)} >= previous\_p_{(i-1)}$, where i refers to the observed number of messages in the conversation, the $confidence$ increases in one (Line 5-6) based on the belief that at some point the prediction will increase sufficiently to exceed the $threshold$. The $threshold$ value was set to 50% because the machine learning classifier considers predictions greater than or equal to 50% as a groomer. So when at some observed message i the $confidence >= min\_confidence$, the mechanism analyzes whether the last prediction exceeded this $threshold$, if so, the mechanism stops reading incoming messages and performs an alert. Specifically, this case is represented in Figure 30a. Note from the figure that the $confidence$ increases because the prediction $new\_p_{(2)}$ is higher than previous prediction $previous\_p_{(1)}$. Eventually, when $confidence = min\_confidence = 3$ at observed message $i = 4$, the mechanism stops waiting for more messages and performs the alert because the last prediction exceeded the $threshold$. If the last prediction had not passed the $threshold$, the mechanism would continue waiting for more information (Line 10-12). The second way for the local $confidence$ value to increase is when the value of $new\_p_{(i)} < previous\_p_{(i-1)}$, but, it is equal to or greater than the $threshold$ (Line 5-6), being the $confidence$ value accepted and increased, refer to Figure 30b for a better illustration of this case. If instead, $new\_p_{(i)} < previous\_p_{(i-1)}$ (Line16-19) and its value is less than $threshold$, the $confidence$ value is reset, i.e., $confidence = 0$ and the analysis starts again. Figure 30c illustrates this case. It can be seen how at message 3 the $confidence$ value is reset to 0 and does not increase again until message 5 when $new\_p_{(5)} > previous\_p_{(4)}$. For this specific case, the penalty is higher because it took more messages to perform the alert, assuming it is a groomer conversion. Finally, there are cases where the $confidence$ value never reaches the desired minimum, refer Figure 30d. When this occurs, the triggering mechanism continues monitoring the system waiting for a change in the predictions. As soon as this does

Algorithm 3 – Triggering mechanism logic structure

```
Require:
    $new\_p_{(i)}$: most recent prediction based on current message $i$;
    $previous\_p_{(i-1)}$: previous prediction based on message $i-1$;
    $delay$: number of messages observed by the preventive system so far;
Ensure:
    $early\_stopping$: boolean value that confirms if an alert must be performed;
    $delay$: value to perform the corresponded penalization.
Set empirically the $min\_confidence$ value;
if $min\_confidence == 1$ then
    return $early\_stopping = 1$, $delay = 1$; // no analysis performed
end if
if $new\_p_{(i)} >= previous\_p_{(i-1)}$ or $new\_p_{(i)} >= 50\%$ then
    $confidence += 1$;
    if $confidence >= min\_confidence$ then
        if $new\_p_{(i)} >= 50\%$ then
            return $early\_stopping = 1$, $delay$;
        else
            return $early\_stopping = 0$, $delay = \infty$; // wait for more information
        end if
    else
        return $early\_stopping = 0$, $delay = \infty$; // wait for more information
    end if
else
    $confidence = 0$
    return $early\_stopping = 0$, $delay = \infty$; // wait for more information
end if
```

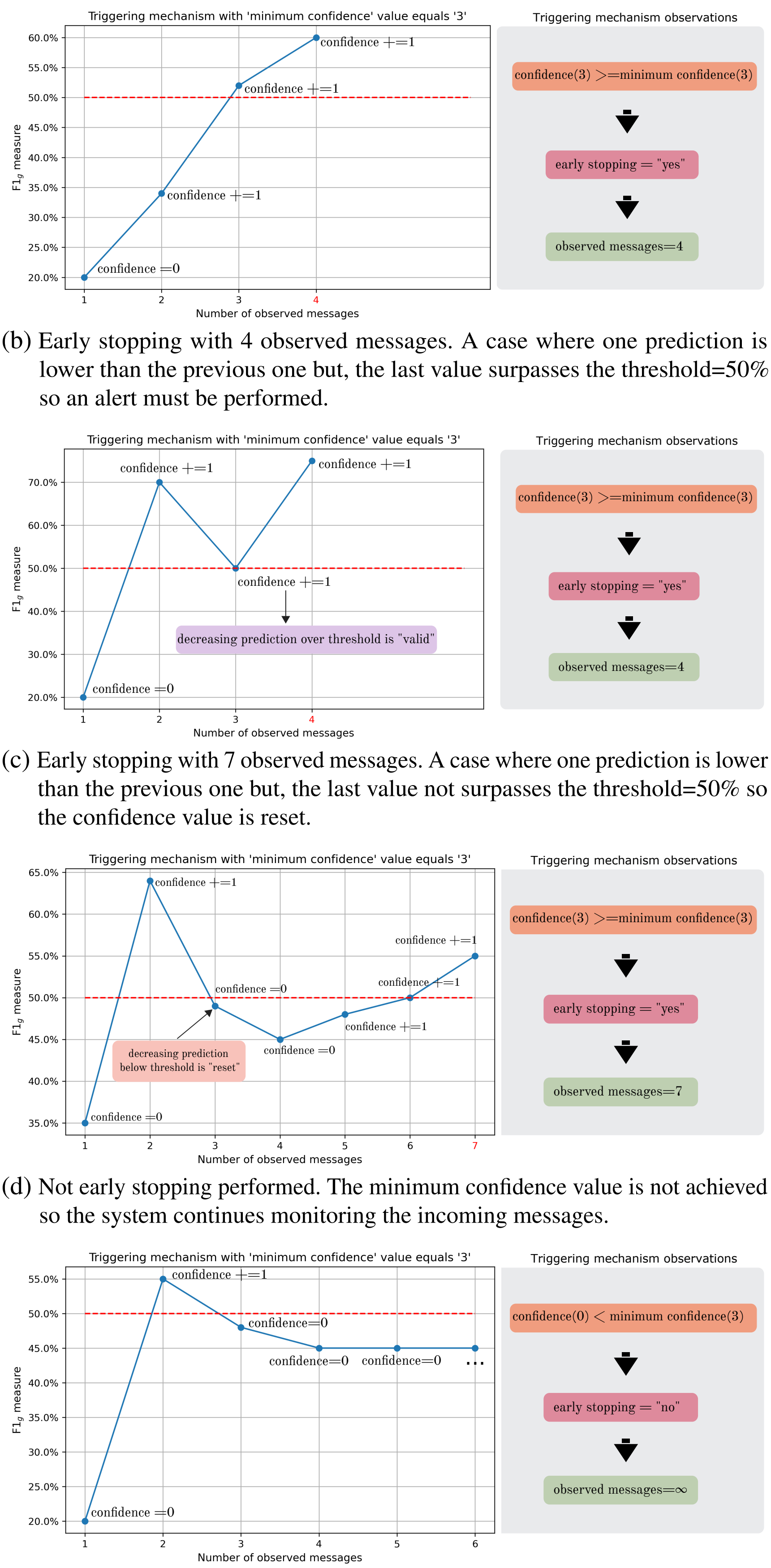
not happen, no alert will be made.

A.4 Evaluating the effectiveness of our proposal

In this section, we evaluate the effectiveness of our triggering mechanism. Recall that the main objective is to perform an alert with as few messages as possible. For this, the triggering mechanism needs to analyze how reliable the preventive system predictions are, i.e., the BF-PSR framework. Furthermore, to evaluate the system in a more realistic environment, the SGD testing corpus is released in a message-by-message format as in real applications. And, we employ the most appropriate evaluation metrics such as latency$_{TP}$, *speed* and F$_{latency}$ in conjunction with F1$_g$ measure to give a complete picture of the triggering mechanism performance.

We start by empirically testing several values of the $min\_confidence$ parameter until we found the ideal one. In Figure 31, the impact of the $min\_confidence$ value on the F1$_g$ and F$_{latency}$ metrics is illustrated. When $min\_confidence = 1$, the mechanism does not perform any analysis and returns the BF-PSR prediction with only one message. This causes that the value of F1$_g$=39% and F$_{latency}$ = 39% to be very low. For a more detailed analysis of this experiment

Figure 30 – Evaluating the possibility of early stopping with our proposed triggering mechanism.



(a) Early stopping with 4 observed messages. A case where each new prediction is greater than the previous one and the last value surpasses the threshold=50% so an alert must be performed.

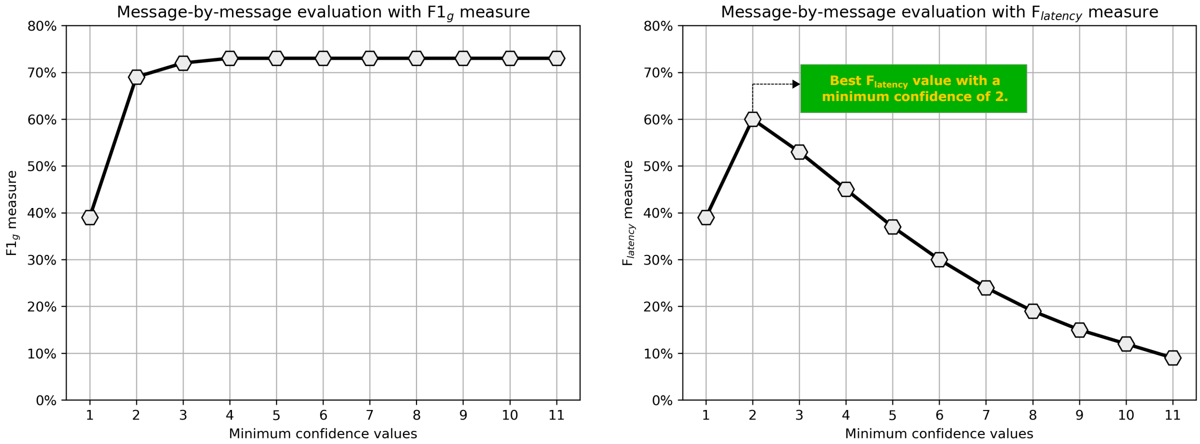
(b) Early stopping with 4 observed messages. A case where one prediction is lower than the previous one but, the last value surpasses the threshold=50% so an alert must be performed.

(c) Early stopping with 7 observed messages. A case where one prediction is lower than the previous one but, the last value not surpasses the threshold=50% so the confidence value is reset.

(d) Not early stopping performed. The minimum confidence value is not achieved so the system continues monitoring the incoming messages.

Triggering mechanism with 'minimum confidence' value equals '3'
Triggering mechanism observations
$F1_g$ measure
Number of observed messages
60.0%
55.0%
50.0%
45.0%
40.0%
35.0%
30.0%
25.0%
20.0%
1
2
3
4
confidence $=0$
confidence $+=1$
confidence $+=1$
confidence $+=1$
confidence(3) $>=$minimum confidence(3)
early stopping = "yes"
observed messages=4
Triggering mechanism with 'minimum confidence' value equals '3'
Triggering mechanism observations
70.0%
60.0%
50.0%
40.0%
30.0%
20.0%
confidence $+=1$
confidence $+=1$
confidence $+=1$
confidence $=0$
decreasing prediction over threshold is "valid"
confidence(3) $>=$minimum confidence(3)
early stopping = "yes"
observed messages=4
Triggering mechanism with 'minimum confidence' value equals '3'
Triggering mechanism observations
65.0%
60.0%
55.0%
50.0%
45.0%
40.0%
35.0%
1
2
3
4
5
6
7
confidence $+=1$
confidence $+=1$
confidence $=0$
confidence $+=1$
confidence $+=1$
confidence $=0$
confidence $=0$
decreasing prediction below threshold is "reset"
confidence(3) $>=$minimum confidence(3)
early stopping = "yes"
observed messages=7
Triggering mechanism with 'minimum confidence' value equals '3'
Triggering mechanism observations
55.0%
50.0%
45.0%
40.0%
35.0%
30.0%
25.0%
20.0%
1
2
3
4
5
6
confidence $+=1$
confidence$=0$
confidence$=0$
confidence$=0$
...
confidence $=0$
confidence(0) $<$ minimum confidence(3)
early stopping = "no"
observed messages=∞

Figure 31 – Triggering mechanism in conjunction with BF-PSR framework results with various $min\_confidence$ values. The plots depict its impact in the F1_g and F$_{latency}$ measures.



refer to Table 19. We can observe that the *speed* of the system is very fast reaching its highest value, i.e., *speed*=1. However, the system has so little information that it is deficient. If instead, the value of $min\_confidence = 2$, a peak occurs in the F$_{latency}$ graph. This means that with a median of two messages in the TP conversations, i.e., latency$_{TP} = 2$, the system has enough information when predicting the scores and reaches its best F$_{latency}$ value; besides presenting a very appropriate system speed, i.e., *speed*=0.87, which is of utmost importance in preventive systems. On the other hand, let us recall that by using all the information available in the testing set, the BF-PSR framework achieves an F1_g of 73% as the best result. So a F1_g of 69% with a $min\_confidence = 2$ is a desirable result for the system. Then, from the figure, we can observe that there is no impact on the effectiveness with $min\_confidence$ values higher than 2. For example, the F1_g scores start to stabilize with a value of 73% and, the F$_{latency}$ values tend to decrease. This occurs because the more messages are required, the higher is the penalization. Recall that if the system needs more than 5 messages as a median, the penalty will be more severe and will approach 1. Now, let us analyze the results of Table 19 in more detail. The best F$_{latency}$ achieved is with $min\_confidence = 2$. If we compare the results with $min\_confidence = 3$, we observe that the F1_g score does not vary much, with a difference of only 4%. However, there is a greater impact on the speed of the system where the speed decrease from 0.87 to 0.74. Thus, the F$_{latency}$ metric prioritizes, for this specific case, the speed of the system rather than the F1_g score. Continuing with the analysis, the worst value of F$_{latency}$ is 0%. This occurs when the system is forced to wait for a large number of messages, e.g., 100 messages, to perform an alert. When it happens, the system speed tends to be 0.00 since it takes a median of about 25 messages to detect the TP cases, i.e, $25 >> median_{TP} = 5$. Of course, as all the information is used, the F1_g measure is 73%. Therefore, in this monograph, we suggest using a value of 2 for the parameter $min\_confidence$.

Table 19 – Metric evaluation results of our proposed triggering mechanism in conjunction with the BF-PSR framework. The results indicate using $min\_confidence = 2$ to achieve a better performance in the mechanism.

$min\_confidence$	$\mathbf{F1}_g$	$\textbf{latency}_{TP}$	$speed$	$\mathbf{F}_{latency}$
1	39%	1	1	39%
2	69%	2	0.87	60%
3	72%	3	0.74	53%
4	73%	4	0.62	45%
100	73%	24.50	0.00	0%

A.5 Summary and discussion

In this appendix, we complemented our preventive system, the BF-PSR framework with a triggering mechanism that allows us to evaluate the effectiveness of the system in a more realistic environment. Since we propose a simple but effective triggering mechanism that decides "when" the predictions are reliable enough to make an alert, new metrics were needed to perform a complete evaluation. Thus, the metrics latency_{TP} and $\text{F}_{latency}$ were employed in the evaluation. From the results obtained, we confirm that our triggering mechanism can perform reliable alerts with a median of two messages per groomer conversation. In other words, the mechanism fulfills the objective of alerting about a possible grooming attack with the least amount of text possible. This was achieved by empirically assigning a value of 2 to the $min\_confidence$ parameter. With it, we achieved a F1_g of 69%, a $speed$ of 0.87, and a $\text{F}_{latency}$ of 60%. We consider the achieved F1_g score to be a positive result since it only differs by 4% when the BF-PSR system uses the **complete** SGD testing set, i.e., 73%. The $\text{F}_{latency}$ and the $speed$ achieved are also very acceptable results that balance the F1_g score with the penalties of late detection of TP cases.

To conclude, we believe that the new triggering mechanism is a valuable contribution since it provides positive results and, to the best of our knowledge, we are the **first** ones to apply a triggering mechanism and its appropriate metrics in the area of early detection of grooming.

ICMC USP

SÃO CARLOS