Online Grooming Detection

Philippe Schläpfer
ETH Zürich

Saahiti Prayaga
ETH Zürich

Simon Iyamu Perisanidis
ETH Zürich

Abstract

Online grooming approaches, where a sexual predator approaches minors online with the goal of sexual abuse, are a big problem in today's world of social media. In this work, we present two approaches to detect sexual predators in chats. We utilize the currently available datasets for *Sexual Predator Detection* (SPD) and analyze their strengths and weaknesses critically. Using dictionary-based and transformer-based approaches, we analyze the writing styles of predators in comparison to non-predators in order to shed light to their differences. Finally, we present our two approaches, one of which improves the current state-of-the-art score by 7.7%. Both approaches are based on BERT models using additional features of the chats as inputs. [1]

1 Introduction

The rise of social media has transformed the way we communicate with each other. Apart from its advantages, social media platforms have given people with bad intentions the ability to engage with potential victims with increased anonymity (Henshaw et al., 2020). Worryingly, there has been a rise in reports of children being exposed to conversations with sexual predators, who try to build trust over longer periods of time in an attempt to prepare the child for sexual abuse (Greene-Colozzi et al., 2020). This phenomenon is called online grooming.

Online grooming is a term used to describe the tactics abusers deploy through the internet to sexually exploit children (Wachs et al., 2012). Solutions for automatically identifying grooming behavior in chats already exist and are used during legal prosecution. However, one caveat of such a solution is that it merely analyzes the incidents after they have happened, instead of being able to prevent them in the first place. Another one is that predictions still have room to improve on accuracy.

Detecting online sexual predators is a challenging task, for a variety of reasons: For one, due to the anonymous nature of chat forums, it is hard to assess the age of the participants in a conversation, especially since most predators attempt to pass as young children and imitate their texting style. Even if a conversation contains sexual content, it is hard to figure out whether it is between two consensual adults or not.

Furthermore, a chat is usually non-contiguous and spans over multiple weeks. Usually a predator tries to build trust and exchange personal information in the first phase, followed by persistent focus on further developing trust. In a third phase, they tend to desensitize victims to sexual topics and isolate them from their parents or support network. They do this through gaslighting and by making them feel like they are the only ones who understand them. In the last phase, they try to arrange meetings. Due to this lengthy and manipulative nature of the grooming process, it is hard to predict predatory behavior, since not a lot of sexual content is being discussed.

In this work, we will attempt to develop and improve upon machine learning models that can detect grooming.

2 Related Work

The main paper whose results we draw inspiration from and seek to improve is Early Detection of Sexual Predators in Chats (Vogt et al., 2021). The paper focuses on prevention of sexual predators by analyzing the level of risk after every message in a running conversation, and triggering an alert after the risk-threshold has been passed.
The paper first formally describes the problem of early sexual predator detection (eSPD) and how to evaluate it. Furthermore, it defines discriminative

[1] The code for the experiments can be found here: https://github.com/schlaepf/CS4NLP_eSPD

features that hint at a grooming attempt, such as talking about age difference, asking about a user's relationship with their parents, etc. It then elaborates on its efforts to assemble the PANC (PAN CLEF 2012, n.d.) dataset (as described in 4.4), and presents a baseline built on BERT-based language models: Messages from an ongoing chat are parsed using sliding windows, output binary predictions, are evaluated using different BERT models, and continuously classified. Sequences of recent window classifications are then compared to the predefined risk-threshold to determine whether it is time to raise an alarm or not. The paper then replicates approaches from related work in similar settings, and it turns out that all the BERT-based language models outperform conventional Sexual Predator Detection (SPD) approaches using SOTA on the VTPAN dataset. The authors of the paper further discuss issues that would need to be addressed in future work, such as the fact that their corpus is from 2012 and does not accurately represent conversations anymore, as the language of the youth has changed rapidly over the years.

The second paper we refer to is "Detecting sexual predators in chats using behavioral features and imbalanced learning" (Cardei and Rebedea, 2017), which treats Sexual Predator Identification as a supervised Machine Learning task with feature selection using data from the PAN 2012 competition (Juola et al., 2012). It first uses a Support Vector Machine (SVM) classifier to filter conversations with predatory undertones from harmless ones. It then uses a Random Forest Classifier (RFC) to determine which of the participants in the conversation is the actual predator. The former classifier, SVM, together with sampling and cost-sensitive methods, is useful for tackling the unbalanced nature of SPD datasets, which are made to reflect the disproportionately low amount of predatory content in real-life conversations. The latter, RFC, was chosen to identify a predatory participant with high accuracy and a very low false-positive rate. The paper deems this system sufficient, since the task of examining a conversation for its predatory part can be manually done by a parent or figure of authority. Both classifiers categorize text using the Bag Of Words (BoW) model, together with some empirical features that are used to quantify the behavior of a predator, such as the number of questions asked by someone, or the percentage of slang/sexual words used. Using feature selection, the most relevant words in the corpus are selected and used as additional features to improve the detection of inappropriate behavior. The paper further suspects that future work could improve their results by adding features which could reflect the sentiment of each participant.

The third paper we draw inspiration from is "On Preprocessing the Data for Improving Sexual Predator Detection" (Borj et al., 2020), which studies various feature-extraction algorithms in order to improve predator identification and the classification of their messages. Both various BoW features from the PAN 2012 are investigated, as well as the GloVe feature set for word embedding features.

3 Problem Statement

The task of "sexual predator detection" (SPD) can be seen as a binary classification problem, where the goal is to classify a given chat of arbitrary length into one of two classes. The classes are 0 (non-predator) or 1 (predator). Note that it's not the goal to identify which chat participant is the predator. We are only interested in finding out if one of the participants is a predator or no participant at all. Also, we define a chat to have exactly two participants, since all the datasets also have that format.
Additionally, it's important that for this paper we do not make the classification on the message level, but on the chat level. Even though this could also be an interesting task, there might occur problems when annotating the data, since it's not clear which message "starts" the grooming attempt.

There is a very important tradeoff to consider between earliness and accuracy. Both these characteristics are not only desirable but also necessary for a SPD system. SPD alerts should ideally be triggered before the last phase of grooming, where the predator can already arrange meetings with their victim. Furthermore, false negatives should be avoided at all costs, since they could lead to sexual assault. False positives are important to avoid as well, since a SPD alert would trigger and involve police action.

4 Datasets

The nature of the task makes it self-evident that, due to legality and privacy, it is difficult to get access to the data. Hence, all the datasets we have used are not publicly available. We have had to

request them from various sources.
In this section we introduce the different datasets known for this task. [2]

4.1 PAN12

The PAN12 dataset was introduced at a CLEF Shared Task from 2012 (Inches and Crestani, 2012). It contains a total of 222k segments of chats of which 2.58% are grooming chats. Segments are parts of chats where a conversation was interrupted for more than 25 minutes. The non-grooming chats for this dataset are sampled from logs of IRC channels from Omegle (Omegle, 2022), while the grooming segments are from decoy operations (not with actual victims). This dataset also contains segments of sexual conversations between consenting adults, which makes it more difficult to predict. Hence, the classifier cannot just focus on sexual words to classify grooming approaches. The goal of this shared task was to classify whether chats are from sexual predators or not. [3]

4.2 VTPAN

The VTPAN dataset is a filtered version of PAN12. It is created by removing low-quality chats from PAN12 (chats that only have 1 participant, chats with less than 6 messages or chats with long sequences of special characters).

4.3 ChatCoder2

The Chatcoder2 dataset contains 497 complete predator chats from the Perverted Justice Foundation Homepage (Foundation, 2004). It was created by (McGhee et al., 2011) and it's worth noting that these chats are real chats with actual victims (unlike the PAN12 dataset). The chats in this dataset are all positive samples, i.e. only chats with grooming attempts.

4.4 PANC

The PANC dataset was created by (Vogt et al., 2021). To evaluate an eSPD system, a dataset in which each segment is annotated as grooming or not is needed. All the existing datasets suffer from the problem that they either consist of unordered segments or they contain only positive samples. This is where the PANC dataset comes in. The authors of the paper "Early Detection of Sexual Predators in Chats" (Vogt et al., 2021) carefully chose samples from both the PAN12 as well as the Chatcoder2 dataset. More precisely, PANC combines all the positive full length chats from Chatcoder2 with the negative segments of PAN12. It then filters out all segments which are shorter than 6 messages and all segments which are longer than 150 messages.
This results in a dataset consisting of $32'510$ segments of which 9.78% are positive segments. Hence, we see that this dataset is very unbalanced. The full statistics of the dataset can be found in (Vogt et al., 2021). This dataset makes it possible to evaluate the earliness of the predictions. The authors describe PANC as the "first corpus suitable for realistic eSPD evaluations".

4.5 Limitations

A possible limitation of the PANC dataset is that it was created by combining two different datasets. Not only that, but the datasets were exclusively used for one label each. The Chatcoder 2 dataset was used for the positive labeled chats only while the PAN12 dataset was used for the negative samples. These two datasets come from two different distributions. It could be that, for example, the writing style of the PAN12 dataset is completely different than the one from Chatcoder 2. There exists a chance that a language model like BERT can distinguish between the different writing styles of the datasets. Therefore, it might be the case that it is too easy for the model to learn the decision boundary for these two classes, not because the model is good at detecting sexual predators, but because these two datasets were generated in different ways.

Another limitation of the datasets is a rather practical one. It stems from the fact that the language of adolescent people changes very quickly. Hence, for practical applications, these datasets may become (or already are) outdated, and therefore the models trained on that data may perform worse in a practice.

5 Data Analysis

With aim to explore the data, and understand the writing style of predators, we employed several data analysis tests. First, we came up with several metrics that could assist us towards this goal. Sec-

[2] A good overview on how to create the different datasets can be found here: https://gitlab.com/early-sexual-predator-detection/eSPD-datasets

[3] Access to this dataset can be requested here: https://zenodo.org/record/3713280

ond, we used XGBoost to determine which ones are the most important for our classification task.

The metrics are hand-crafted features, Emotion BERT scores and LIWC dictionary categories. Namely, we crafted the following features: message length, size of words, number of questions asked. Emotion BERT is trained on the GoEmotion dataset and for every sentence that is given as inpute it outputs a score for every emotion. For example anger, fear, confusion, disappointement etc. LIWC is a dictionary-based tool for analyzing text across lexical categories. Using LIWC to analyze language can help understand their thoughts, feelings, personality, and the ways they connect with others. Some of the 193 categories are: sexual, love, family, work, wedding etc.

For each metric, we tried to distinguish how the writing style of predators is different than non-predators. We did this by looking at the relative difference of each score between the two participants of the conversation. We found that the LIWC categories that are the most distinct for predators are: help, office, dance, money, wedding, domestic_work, sleep, medical_emergency, cold, hate, cheerfulness, aggression, occupation, envy, anticipation.

In table 1 the full list of the emotion labels and the output of the fine-tuned BERT model applied to the training split of the PANC dataset is shown. The scores are aggregated using the median. Noticeable is the difference of non-predator and predator for the emotion 'amusement' and for 'optimism'.

XGBoost enabled us to depict the feature importance for every feature that we extracted. This is displayed in Figure 1.

6 Reproducing results

The baseline method consists of two stages. As a preprocessing step, the conversations are split into sliding windows and tokenized. The first stage involves doing binary classification for each window using BERT. During the second stage, as illustrated in Figure 2, we can view the output of the binary labels as a sequence of zeros and ones, which are compared to the predefined risk-threshold to determine whether the conversation contains too much predatory content or not. This approach benefits the early detection of sexual predators, since one can continuously analyze the level of risk after every window, accumulate it according to a predefined threshold, and trigger an alert once the threshold

label	non-predator	predator
admiration	0.034316	0.022022
amusement	0.162595	0.284108
anger	0.035295	0.015832
annoyance	0.057766	0.038928
approval	0.024273	0.024937
caring	0.008864	0.014139
confusion	0.07367	0.059846
curiosity	0.194928	0.159221
desire	0.004637	0.015174
disappointment	0.015393	0.019721
disapproval	0.020591	0.01957
disgust	0.009837	0.007383
embarrassment	0.004542	0.005942
excitement	0.009211	0.008643
fear	0.002083	0.003463
gratitude	0.02213	0.007749
grief	0.00087	0.001452
joy	0.026671	0.039703
love	0.016361	0.036839
nervousness	0.00226	0.003216
optimism	0.011192	0.040347
pride	0.002616	0.002325
realization	0.015647	0.016026
relief	0.00223	0.00181
remorse	0.003874	0.011506
sadness	0.008249	0.018981
surprise	0.00896	0.004941
neutral	0.220939	0.116173

Table 1: The median scores of the GoEmotion BERT model applied to the PANC dataset.

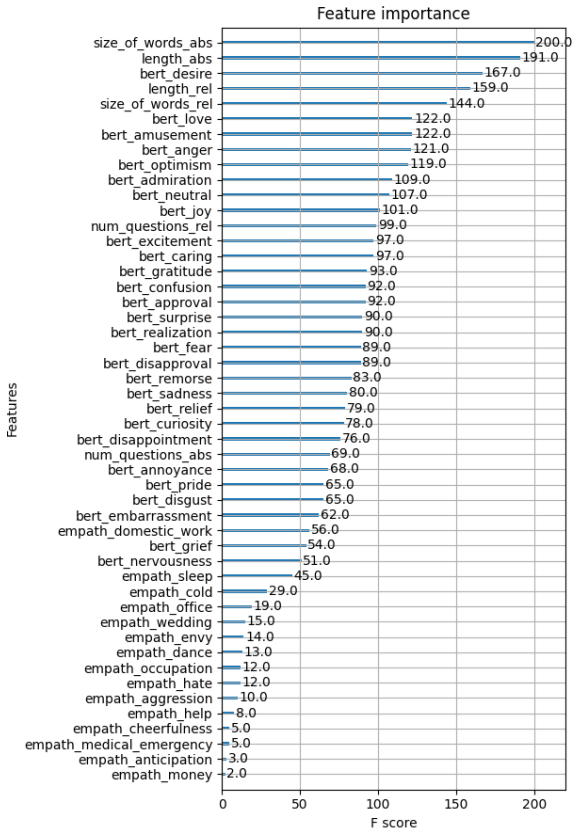

Figure 1: XGBoost feature importance in determining which conversations are predatory

has been passed. In their work, Vogt et al. simply used a constant threshold, i.e a chat segment was classified as predatory if at least five windows were classified as such.

The downside of this approach is that conversation history is not taken into account. In order to alleviate this problem, we experimented with more complex models involving Time Series Classification of the output predictions of BERT (instead of simple thresholding), in the hope of capturing conversation history and making a more informed classification. However, we quickly noticed that, due to the discrepancy in the amount of messages each segment of our data contains (ranging anywhere from 6 messages to 150), which resulted in short, variable-length time series, our predictions did not improve.

Therefore, we opted to make a trade-off in favor of accuracy as opposed to earliness of predictions, and spent our remaining time trying to improve the detection of predators in entire conversation segments, as opposed to real-time analysis of conversations as they progress.

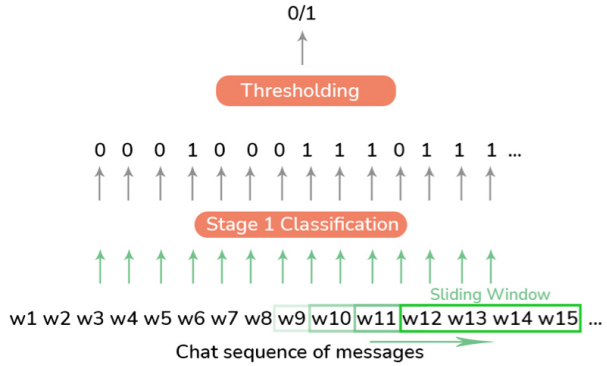

Figure 2: BERT + Sliding Window Classification

7 Approach

Our approach to improving the accuracy of sexual predator detection consists of extracting features of the conversations and using a fusion of different models to make the final prediction. Based on the characteristics that we found distinctive in our data analysis, we extract useful features. In our first attempt, we use BERT, concatenate the features with the CLS Embedding of BERT and use an MLP in the end of the architecture. In our second approach, which produces results better than the current state-of-the-art, we ensemble BERT and XGBoost (Chen and Guestrin, 2016) by aggregating their predictions.

7.1 Preprocessing and Feature Extraction

As a preprocessing step, we use the tokenizer of the Hugging Face library (Wolf et al., 2020). This step includes splitting strings in sub-word token strings, converting tokens strings to ids and back, and encoding/decoding (i.e., tokenizing and converting to integers). We experimented with adding special tokens between the messages depending whether the sender changed, in order to better distinguish between who sends which message, but it did not improve our results.

As discussed in the Data Analysis segment, we tested our hypothesis that some features can help us distinguish between the writing styles of predatory vs non-predatory conversations. We saw that indeed there is an important difference in the writing style of the two classes. Namely, the features that we extracted were: Handcrafted Features (message length, size of words, number of questions asked), Emotion BERT scores, LIWC (Pennebaker) categories. For each feature, we use it's absolute value, but also the relative difference of this value between

the two participants of the conversation in order to calculate how different the writing styles of the two senders are. We added those features to our training dataset, in order to increase our predictions.

7.1.1 BERT fine-tuned on GoEmotions dataset

To extract additional features, a BERT (Devlin et al., 2018) model that is fine-tuned on the GoEmotions dataset (Demszky et al., 2020) was used. This dataset contains 58k English Reddit comments, which are labeled for 27 emotion categories or neutral. Amongst the categories are admiration, disapproval, excitement, etc. There was no fine-tuning done with this model.
For each chat, the BERT model was used to assign scores denoting how likely a chat represents an emotion. So we ended up with 28 additional features for each chat. The full list of emotion labels as well as results and statistics of the BERT model applied to the PANC dataset can be found in the data analysis section (5).

7.2 LIWC

LIWC is a tool for analyzing text across lexical categories. Using LIWC to analyze language can help understand their thoughts, feelings, personality, and the ways they connect with others. Because LIWC requires a paid licence, we used an open-source equivalent which is called EMPATH (Fast et al., 2016) instead. We selected 15 categories which we found are the most distict between the two labels. Those are: help, office, dance, money, wedding, domestic_work, sleep, medical_emergency, cold, hate, cheerfulness, aggression, occupation, envy, anticipation.

7.3 Approach 1: Adding features to BERT's CLS Embedding

BERT is very skilled at making sense of language, but it struggles to understand the relative values of numbers. Since the features we extracted are all numerical, it made little sense to pass them to our BERT model as well. There would even be a chance that one could run into problems with BERT's 512 token limit. Furthermore, since BERT is a powerful model capable of encoding everything needed for classification into its [CLS] embeddings, it just uses a simple linear classifier on the output.

Therefore, since we have some additional features outside of BERT, we needed a more complex model on the output capable of combining the different sources of information intelligently.
In our first attempt to include the features we extracted, we concatenated them with the CLS embeddings of our original BERT model and used a simple Multi-Layer Neural Network, which took this vector as input to make the final prediction of whether this segment contained predatory behavior or not (see 3).

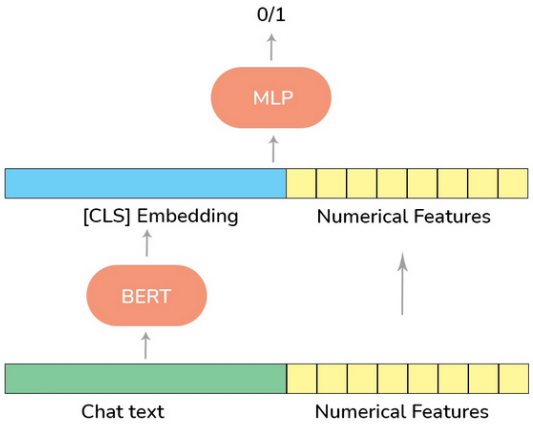

Figure 3: Approach 1: Concatenating features with BERT embeddings and passing them to an MLP

This approach was taken from the Multimodal Toolkit (Gu and Budhkar, 2021), which is an open-source Python package that helps to incorporate categorical and numerical features with Transformer architectures. The implementation of the training loop was heavily inspired by Chris McCormick's research series on BERT + Categorical Features (McCormick, 2021b).

Following (McCormick, 2021b)'s approach, we first normalized our numerical features, since it is easier for the MLP classifier to deal with data of the same distribution. Then, chat segments were preprocessed according to the methods in section 7.1.

The implementation of the MLP as well as the hyperparameters for the model were taken directly from the Multimodal Toolkit (Gu and Budhkar, 2021), as, after experimenting with our own simple MLP and parameters, the ones from the toolkit turned out to have the best performance.

7.4 Approach 2: Ensemble BERT and XGBoost predictions

After having identified that our features can describe the writing style of predators pretty well, we attempted using simple machine learning to classify

the chat segments. That is, using the feature alone, without including the text of the messages. We experimented with XGBoost because of how efficient it has shown to be in similar scenarios. Training such a model only takes a couple of seconds. Surprisingly, as we will see in the next section, the results are very high.

Lastly, we experimented with ensembling two methods (Figure 4, in order to get best of both worlds. We aggregated the raw probability predictions of BERT and XGBoost by averaging them. This enabled us to achieve our highest f1 score.

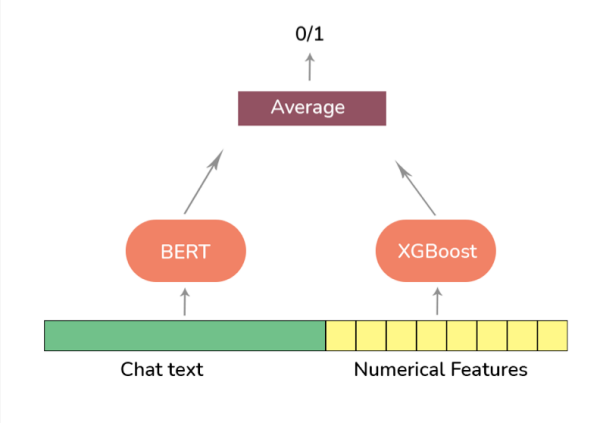

Figure 4: Ensemble of BERT and XGBoost

8 Evaluation

Approach	F1	Precision	Recall
Vogt et al.	0.89	0.82	0.96
BERT	0.968	0.971	0.965
BERT + MLP	0.65	0.62	0.69
XGBoost	0.967	0.948	0.987
BERT + XGBoost	0.967	0.989	0.998

Table 2: The results of the different approaches on the PANC dataset.

In table 2 all the results are listed. "Vogt et al." denotes the results of the previously mentioned paper. One can notice that except for the "BERT + MLP" approach, all of our approaches score well above 0.9 F1.

9 Discussion

As one can discern from the table, our first approach (BERT+MLP) has the worst performance. The F1 score used to be worse until we added batch normalization on the numerical features, which is a technique the Multimodal Toolkit (Gu and Budhkar, 2021) recommends, since it is supposed to stabilize features as they pass through the model and helps with convergence.

Our first suspicion was that introducing too many additional features might have been counterproductive as it may have acted as noise, but this suspicion was resolved when we saw how high XGBoost was able to score when fed with the features alone. We suspect one reason for the subpar performance is that the model we used (in particular the MLP) is not complex enough to handle the massive class imbalance present in the dataset. Perhaps stratification could be a solution for this. Analyzing the MCC score and the amount of true/false positives/negatives also shows that the model is practically incapable of detecting the few true positives present, which is detrimental to the task of SPD. We are yet to find out why combining the two approaches results in a lower score.

It is worth mentioning that the Multimodal toolkit consists of multiple classes, and together, these classes implement about 6 different strategies for combining the features. We oriented ourselves around McCormick's flattened-down design of the class that supports concatenation. Perhaps experimenting with other implementations would result in a higher performance.

Another concern that we have, as mentioned in the sections about the limitations of the datasets, is how robust our models are to real-life scenarios. We can not be certain that our model is actually good at detecting predators, and that it's not just performing well because of the way the dataset is generated. It would be insightful to test our model in different datasets in the future.

10 Conclusion

In conclusion, we have used currently available datasets to perform grooming detection in online chats. These datasets are a step in the right direction, but suffer from severe limitations. We have analyzed the differences between predators and non-predators in chats, and built an intuition of what distinguishes them. Using hand-crafted features, Emotion BERT scores and LIWC categories we have supplemented our classification pipelines to achieve better results. For the classification of the chats, we experimented with two approaches. One is concatenating the features to the CLS embedding of BERT, and the other is ensembling BERT

and XGBoost. The results we achieved are better than the current state of the arts. However there is further research that needs to be done before such models can be trustworthy enough to be used in a real-life scenario.

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Parisa Rezaee Borj, Kiran Raja, and Patrick Bours. 2020. On preprocessing the data for improving sexual predator detection : Anonymous for review. In *2020 15th International Workshop on Semantic and Social Media Adaptation and Personalization (SMA*, pages 1–6.

Claudia Cardei and Traian Rebedea. 2017. Detecting sexual predators in chats using behavioral features and imbalanced learning. *Natural Language Engineering*, 23(4):589–616.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. Goemotions: A dataset of fine-grained emotions.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Ethan Fast, Binbin Chen, and Michael S. Bernstein. 2016. Empath: Understanding topic signals in large-scale text. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 4647–4657, New York, NY, USA. Association for Computing Machinery.

Perverted Justice Foundation. 2004. Perverted justice foundation homepage.

Emily A. Greene-Colozzi, Georgia M. Winters, Brandy Blasko, and Elizabeth L. Jeglic. 2020. Experiences and perceptions of online sexual solicitation and grooming of minors: A retrospective report. *Journal of Child Sexual Abuse*, 29(7):836–854. PMID: 33017275.

Ken Gu and Akshay Budhkar. 2021. A package for learning on tabular and text data with transformers. In *Proceedings of the Third Workshop on Multimodal Artificial Intelligence*, pages 69–73, Mexico City, Mexico. Association for Computational Linguistics.

Marie Henshaw, Rajan Darjee, and Jonathan A. Clough. 2020. Chapter five - online child sexual offending. In India Bryce and Wayne Petherick, editors, *Child Sexual Abuse*, pages 85–108. Academic Press.

Giacomo Inches and Fabio Crestani. 2012. Overview of the International Sexual Predator Identification Competition at PAN-2012. In *CLEF 2012 Evaluation Labs and Workshop – Working Notes Papers, 17-20 September, Rome, Italy*. CEUR-WS.org.

Patrick Juola, Giacomo Inches, Efstathios Stamatatos, Shlomo Argamon, Moshe Koppel, and Fabio Crestani. 2012. Pan lab at clef 2012: Sexual predator identification.

Chris McCormick. 2021a. Combining categorical and numerical features with text in bert.

C.M [ChrisMcCormickAI] McCormick. 2021b. Mixing BERT with Categorical and Numerical Features.

India McGhee, Jennifer Bayzick, April Kontostathis, Lynne Edwards, Alexandra McBride, and Emma Jakubowski. 2011. Learning to identify internet sexual predation. *International Journal of Electronic Commerce*, 15(3):103–122.

Omegle. 2022. Omegle: Talk to strangers!

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

James W Pennebaker. Linguistic inquiry and word count: Liwc 2001.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

Matthias Vogt, Ulf Leser, and Alan Akbik. 2021. Early detection of sexual predators in chats. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4985–4999, Online. Association for Computational Linguistics.

Sebastian Wachs, Karsten D Wolf, and Ching-Ching Pan. 2012. Cybergrooming: risk factors, coping strategies and associations with cyberbullying. *Psicothema*, 24(4):628–633.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Implementation Details for approach 1

As opposed to our original implementation of BERT, which uses the Trainer API Huggingface Transformer's library, the fine-tuning of this approach was done with native PyTorch (Paszke et al., 2019). This is due to the fact that writing the training loop ourselves with PyTorch gave us more freedom to change things. Trainer takes care of everything and allows one to fine-tune a model in a single line of code. By writing the training loop ourselves, it was easier to see how to obtain the CLS embeddings of the model.

Following (McCormick, 2021b)'s approach, we created a custom BERT class, which inherits from transformer's BertForSequenceClassification. By modifying its forward pass method, we allow it to receive the current batch's numerical feature vector as an additional parameter, which can then be concatenated with the CLS token embedding, which is present in the 0th slice of the output tensor of the last layer. This vector is then passed through the MLP, which is initialized upon creation of the custom BERT model. The training loop is the same as any usual PyTorch training loop, except that the forward call is made to the custom BERT class.

The training loop looks as follows (McCormick, 2021a):

1. Unpack data from (train/validation) dataloaders and load into GPU for acceleration

2. Clear gradients calculated in the previous pass, since PyTorch accumulates them by nature

3. Forward pass through the custom BERT model. CLS embeddings are extracted and concatenated with vector of numerical features. Resulting vector is passed through the MLP, loss is calculated and returned.

4. Backpropagation on loss

5. Optimizer updates parameters

Parameters: batch size = 16, learning rate = 3e-3, epochs = 4