Proceedings of the Workshop on Social Threats in Online Conversations: Understanding and Management (STOC-2020), pages 15–20
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Analysis of Online Conversations to Detect Cyberpredators Using Recurrent Neural Networks

Jinhwa Kim, Yoon Jo Kim, Mitra Behzadi, Ian G. Harris
Dankook University, Seoul Women's University, University of California Irvine
Yongin South Korea, Seoul South Korea, Irvine CA USA
best08618@gmail.com, yoonjokim12@gmail.com, mbehzadi@uci.edu, harris@ics.uci.edu

Abstract

We present an automated approach to analyze the text of an online conversation and determine whether one of the participants is a cyberpredator who is preying on another participant. The task is divided into two stages, 1) the classification of each message, and 2) the classification of the entire conversation. Each stage uses a Recurrent Neural Network (RNN) to perform the classification task.

Keywords: Cyberpredator detection, natural language understanding, recurrent neural networks

1. Introduction

Online cyberpredators are a serious threat against children who increasingly use social networking and messaging services to interact with strangers.Our study also found that one in nine teens will received unwanted online solicitations (Madigan et al., 2018). Parents are advised to monitor their children's use of social media, but this is extremely difficult in practice given the variety of networking services and access methods that a child can choose from. Several software tools are available to observe children's online behavior (FlexiSPY, 2019; Easemon Inc., 2019; Cocospy, 2019). However, existing products are limited to recording data for later examination, or providing a "keyword alert" when a particular word has been used in text. These tools do not attempt to understand the semantics of the conversation, so the majority of the burden of identifying cyberpredators is left to the parent's manual effort. Automated approaches which employ natural language understanding could be of tremendous benefit. We present an approach to automatically monitor communications with a child in order to determine if a communication partner is a cyberpredator.

Machine learning approaches, specifically artificial neural networks (ANNs), are generally well suited to this type of classification problem because they can theoretically approximate continuous functions, given a few assumptions. An ANN could be used to classify a conversation as either predatory or non-predatory, however there are several practical difficulties in the application of ANNs to this problem. One issue is the importance of context in understanding the meaning of a conversation. Attempting to infer the intent of a conversation by examining utterances individually will generally produce poor results because sentences in dialogs are meant to be understood in the context of all messages in the dialog. The dependence on extended context requires that the input to the classification process must be a large block of utterances which must be classified as a whole. Another difficulty is the high dimensionality of the input space of the problem, which must capture entire conversations with hundreds of messages.

We address the high dimensionality by dividing the problem into two stages. The first stage classifies the intent of individual messages, and the second stage uses the results of the first stage to classify the entire conversation. The first stage generates a concise summary of the individual messages, allowing the second stage to efficiently consider the meaning of the entire conversation. We address the context in two ways. When classifying individual messages, the first stage also considers the a window of 5 messages which comprise local context. When classifying the entire conversation, the second stage considers the classifications of all messages uttered by the potential attacker in the conversation.

2. Related Work

Much of the existing research in detection of cyberpredators is based on the chat log transcripts provided by Perverted Justice (Perverted Justice Foundation, 2019), a community of volunteers who posed as children in chat rooms in order to lure predators. The efforts of the Perverted Justice community has been credited with resulting in the conviction of 623 cyber-predators to date. Chats with predators have been transcribed and made available to the public. The linguistic properties of the Perverted Justice dataset have been explored in several studies (Black et al., 2015; Chiu et al., 2018). The International Competition for Sexual Predator Identification was held and the PAN 2012 workshop (Inches and Crestani, 2012b), catalyzing interest in the problem. To support the competition, the PAN 2012 dataset was created using the Perverted Justice dataset and enhancing it with adult-to-adult sexual conversations from a repository of Omegle conversations and a set of IRC chat logs (Inches and Crestani, 2012c).

Almost all existing approaches use machine learning approaches to detect predatory text. Many machine learning techniques have been used including Support Vector Machines (Pendar, 2007; Morris and Hirst, 2012; Parapar et al., 2012; Peersman et al., 2012; Villatoro-Tello et al., 2012; Escalante et al., 2013; Vartapetiance and Gillam, 2014; Cheong et al., 2015), Decision Trees (McGhee et al., 2011a; Miah et al., 2011; Kontostathis et al., 2012; Vartapetiance and Gillam, 2014; Cheong et al., 2015), Naive Bayes (Miah et al., 2011; Bogdanova et al., 2012; Vartapetiance and Gillam, 2014; Cheong et al., 2015), k-Nearest Neighbor

(Pendar, 2007; Cheong et al., 2015), logistic regression (Miah et al., 2011; Cheong et al., 2015), Maximum Entropy (Eriksson and Karlgren, 2012), and Multilayer Perceptron (MLP) Neural Networks (Villatoro-Tello et al., 2012; Escalante et al., 2013; Cheong et al., 2015). A rule-based heuristic was presented (McGhee et al., 2011a; Kontostathis et al., 2012) and shown to outperform a decision tree approach.

All approaches, other than those based on Neural Networks, require the explicit definition of set of features used to represent the conversation. All of these approaches have used lexical features, unigrams and bigrams which are associated with speech acts commonly performed by attackers. Words are grouped into dictionaries which are assumed to indicate the conversational goals of a predator. Examples of lexical features include the number of desensitization verbs (e.g. kiss, suck) and the number of reframing verbs (e.g. teach, practice) (McGhee et al., 2011a; Kontostathis et al., 2012). Several approaches use Linguistic Inquiry and Word Count (LIWC) (Pennebaker et al., 2011) features which associate words with cognitive and emotional states. Some approaches use conversational/behavioral features which model properties of the overall dialog such as the number of conversation participants, the length of the conversation (Eriksson and Karlgren, 2012), the number of initiations, and the number of questions (Morris and Hirst, 2012).

Several previous approaches have employed MLP neural networks to identify predators (Villatoro-Tello et al., 2012; Escalante et al., 2013; Cheong et al., 2015). Neural networks do not require an explicit set of features. Instead, these previous approaches use a bag-of-words representation which summarizes a conversation as the number of occurrences of each word in the vocabulary, regardless of sequence.

3. Cyberpredator Intent Classification

Early work in the study of online child exploitation presented a set of conversational goals of predators and categorized the utterances of a predator based on the goal being achieved. A typology is presented by O'Connell (O'Connell, 2003) which describes 5 stages on conversation: friendship forming, relationship forming, risk assessment, exclusivity, and sexual. An alternate classification is presented by Olson (Olson et al., 2007) which contains 3 main classes: grooming, isolation, and approach.

Researchers in (McGhee et al., 2011a) present a classification of cyberpredator intents and a tool, ChatCoder2, which uses a rule-based approach to classify messages according to their classification. We use the classification presented in (McGhee et al., 2011a) because it has been shown to be effective, and because we can use the ChatCoder2 tool to generate a labeled dataset which we use for training. Each message is placed in one of the following 4 classes.

- **Exchange of personal information (200)** - This includes questions about semi-personal information which might be exchanged between new friends. Topics include age, gender, location, boyfriends/girlfriends, and likes/dislikes. The cyberpredator uses this to initiate a trust relationship.
- **Grooming (600)** - This involves the use of sexual terminology, regardless of context. Cyberpredators often use this to desensitize the victim to sexual discussions.
- **Approach (900)** - This describes when the cyberpredator is either gathering information to arrange a meeting, or encouraging the victim to keep their relationship secret.
- **Non-predatory (000)** - These are all messages not in one of the previous classes.

4. System Architecture

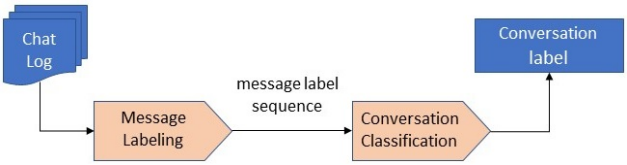

Figure 1: Overall Structure

The overall structure of our approach is illustrated in Figure 1. Our approach is divided into two stages. The first stage is *Message Labeling* which labels each of the message from the potential attacker with its intent classification. The second stage is *Conversation Classification* which evaluates the entire sequence of messages from the potential attacker and determines if the potential attacker is a predator or not. The input to the Conversation Classification stage is not a sequence of messages. Instead, it is the sequence of sentence labels produced by the Message Labeling stage.

Message Labeling During the Message Labeling stage, each message from the potential attacker in a dialogue is categorized by its intent classification. For categorizing each message, we train the mapping pattern between a message and the categories. Because the messages are written in natural language, each message must be converted in to the form of a vector to be used by the training model. In order to capture the meaning of each message, we must consider not only the words in the message and their sequence, but also the preceding messages which comprise the context in the conversation. We use two different methods to represent the meaning of a message, one method to represent meaning in a single message, and a second method to consider the impact of context.

We generate message encodings using the Universal Sentence Encoder (Cer et al., 2018a) approach to generate a message vector. This is in contrast to other traditional methods such as word2vec embeddings (Mikolov et al., 2013) or bag-of-words model. Word embeddings have been shown to perform well at capturing the meaning of individual words. However, the bag-of-words model loses meaning information because it ignores the ordering of words in the message. The Universal Sentence Encoding approach uses an LSTM (Long Short-Term Memory models) model to understand the relationship between each word. For this reason, Universal Sentence Encoding is more suitable and we employ it using its TensorFlow (Cer et al., 2018b) implementation.

In addition, the meaning of a message is determined in part by the context which precedes it. Though two statements look similar, they could have different meanings depends on their preceding contexts. The message "Call me" could be a message from the two close friends while it could be one that a predator leads a victim to call him or her. In this case, it is hard to identify where it would belong to with only one message. For this reason, when classifying a message, we consider the 4 messages preceding the message in question. So classification is performed by examining a window of 5 messages in order to consider conversational context.

Figure 2 is the structure of our training model for Message Labeling. The window of 5 messages, ending with the message being classified, are input to a layer which uses Universal Sentence Encoding to generate a 1*512 dimension vector for each message. The vectors from the encoding layer become the input of the succeeding LSTM and then Dense layer. By considering the window of 5 vectors at the LSTM/Dense layers, our approach can infer local context.

Conversation Classification After labeling each message, each conversation is represented as a sequence of labeled statements. These label values are used as the value of the vector for input to the *Conversation Classification* stage. Figure 3 shows the structure of our model of the Conversation Classification stage. The sequence of message labels is padded to ensure that the length of the input vector is constant. However, the padded labels must not be considered as the labels of the conversation. Therefore, we use the Masking layer, which is used to ignore padded labels. We use an LSTM layer to train the corresponding pattern of labels and then a Dense layer for final classification of the conversation.

5. Experiments

We present two sets of results. The first set of results evaluate the Message Labeling stage alone by presenting the precision and recall of the message labeling process. The second set of results evaluates both the Message Labeling and Conversation Classification stages together by presenting the precision and recall of the classification of a set of conversations.

5.1. Dataset

To evaluate the Message Labeling stage, we use chatlog data from both ChatCoder2 (McGhee et al., 2011b) and PAN2012 (Inches and Crestani, 2012a). ChatCoder2 provides conversations extracted from the Perverted-Justice (PJ) website (Perverted Justice Foundation, 2019). All of the conversations in the ChatCoder2 dataset are predatory, while the PAN2012 dataset if a mix of predatory and non-predatory conversations. ChatCoder2 is an heuristic tool which automatically labels each message with its intent classification ('000', '200', '600', and '900'). We use ChatCoder2 to automatically classify the messages in each conversation.

Table 1 describes the set of messages used to evaluate the Message Labeling stage. A total of 5008 messages are used and are taken from both the ChatCoder2 and PAN2012 datasets. Our goal was to use the same number of messages in each predatory intent ('200', '600', and '900'), so we extracted 626 of each type of message from the ChatCoder2 dataset, and we selected another 626 messages with non-predatory intents ('000') from the ChatCoder2 dataset for balance. The number 626 was chosen because that is the largest number of messages that we could select while maintaining balance in each intent. In other words, 626 is the minimum of the number of messages in each class in the ChatCoder2 dataset. In total, 2504 (626 * 4) messages are selected from the ChatCoder2 dataset. We expect that using non-predatory messages ('000') only from the ChatCoder2 dataset would result in a biased classification because all of the non-predatory sentences would be taken from predatory conversations. For this reason, we selected another 2504 non-predatory messages from the PAN2012 dataset as well.

Total Dataset	Number
♯ of Conversations	119
♯ of Total Messages	5008
♯ of Messages in Category '000'	3130
♯ of Messages in Category '200'	626
♯ of Messages in Category '600'	626
♯ of Messages in Category '900'	626

Table 1: Dataset used to evaluate Message Labeling

Total Dataset	Number
♯ of Conversations	480
♯ of Predatory Conversations	128
♯ of Non-Predatory Conversations	352
♯ of Total Messages	78130

Table 2: Dataset used to evaluate Conversation Classification

To evaluate the Conversation Classification stage, we use only the PAN2012 dataset for training and test. We use only conversations with more than 130 messages. The sequence of labeled messages from the output of Message Labeling are used as input for Conversation Classification. Conversations in the PAN2012 dataset are pre-labeled as predatory and non-predatory. Table 2 describes the properties of the dataset to evaluate Conversation Classification.

5.2. Results of Message Labeling

For training the network used for Message Labeling, we use 10 epochs use a batch size of 32. We use 80% of the dataset for training and 20% for testing. Table 3 shows the precision and recall values for each label, independently. Both training and testing are performed on an Intel Xeon CPU,

	Label	000	200	600	900
Training	Precision	0.93	0.76	0.73	0.68
	Recall	0.91	0.78	0.79	0.65
Test	Precision	0.91	0.77	0.73	0.68
	Recall	0.92	0.78	0.79	0.64

Table 3: Performance results of Message Labeling

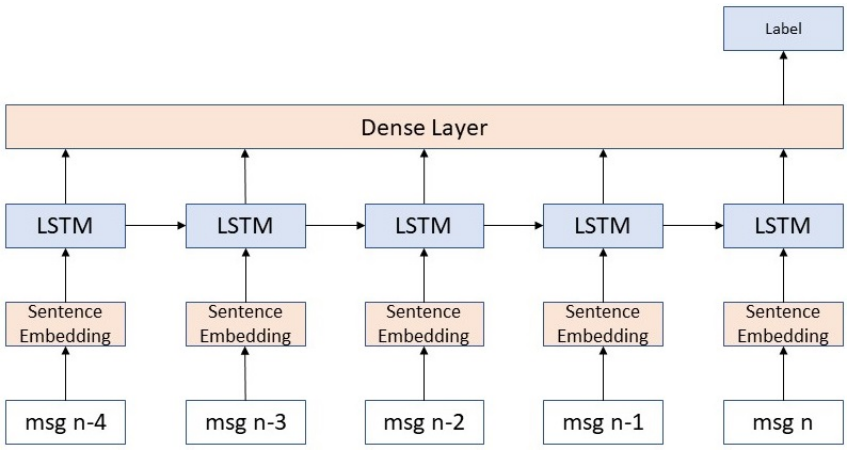

Figure 2: Structure of *Message Labeling*

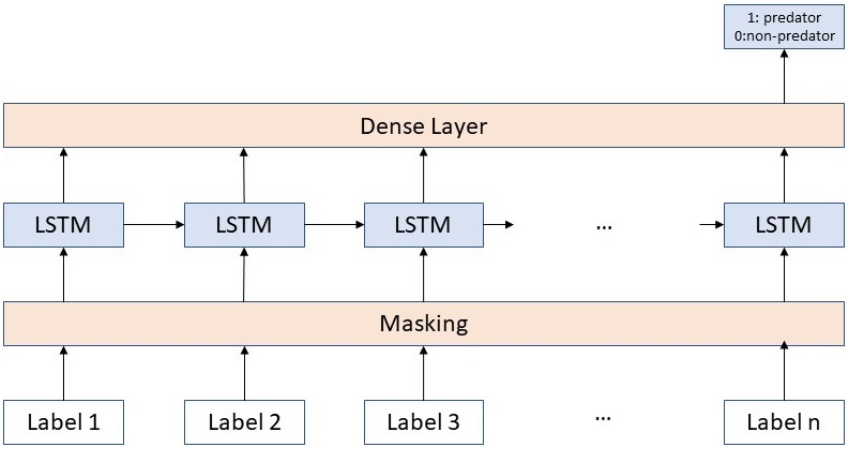

Figure 3: Structure of *Conversation Classification*

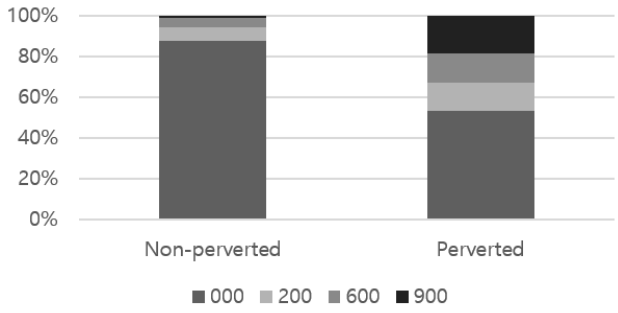

Figure 4: Distributions of each label in conversations from either predator or non-predator

2.3GHz clock rate, with a Tesla K80 GPU, within Google Colaboratory. The entire training process is performed in 1 minute and 12 seconds. Sentence embedding requires 29 seconds of the total time.

5.3. Results of Conversation Classification

Conversation Classification was evaluated by using Message Labeling to label each message in the PAN2012 dataset, and using the resulting message label sequences to classify each conversation. Table 4 shows the number of sentences with each label in the PAN 2012 dataset, as derived using the trained model for Message Labeling. Figure 4 shows the distribution of different message labels in both predatory and non-predatory conversations. In non-predatory conversations, the vast majority of messages, 88%, are classified in set '000' as clearly non-malicious. In predatory conversations, the percentage of '000' messages is lower, 53%, and the other potentially predatory message classes are much more common.

Label	**000**	**200**	**600**	**900**
♯ of messages	59142	7060	6262	5666

Table 4: Number of PAN 2012 messages in each class

Predicted	Actual	
	Predatory	Non-predatory
Predatory	29(TP)	3(FP)
Non-predatory	2(FN)	62(TN)

Table 5: Performance results of categorization conversations

When training the model for Conversation Classification, we set the maximum length of the sequence as 200 and the input whose length is lower than 200 is padded. We use 10

epochs and set batch size to 32. We use 80%(384) of the dataset for training and 20%(96) for the test. Our model yields precision of *0.9063*, recall of *0.9355*, F1 score of *0.9148*, and F0.5 score of *0.9058*. Table 5 shows the detail of the performance results. We compare our results to those presented at the PAN2012 cyberpredator detection competition (Inches and Crestani, 2012b), although our dataset included ChatCoder2 data, in addition to the PAN2012 data used in the competition. Compared to the 16 competitors used for official evaluation, our results place us first with respect to recall, first with respect to F1 score, third with respect to F0.5 score, and fifth with respect to precision. We argue that recall is the most important measure for this problem because it indicates the fraction of predators who would go undetected. We expect that a parent would be more willing to accept a small number of false alarms rather than risking the possibility of missing a predator.

6. Conclusions

We have presented an approach to the detection of predatory conversations which first classifies individual messages and uses those results to classify entire conversations. RNNs are used to perform each stage and are trained using messages labeled by the ChatCoder2 tool and existing prelabeled conversations. Limited context is considered in the labeling of individual messages by considering the previous 4 messages when classifying a message. Our approach provides better recall than previous approaches.

7. Ethical Considerations

Our contribution is focused on helping to protect children from cyberpredators. We do not foresee any malicious use of this technology.

8. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1813858. This research was also supported by a generous gift from the Herman P. & Sophia Taubman Foundation.

9. Bibliographical References

Black, P. J., Wollis, M. A., Woodworth, M., and Hancock, J. T. (2015). A linguistic analysis of grooming strategies of online child sex offenders: Implications for our understanding of predatory sexual behavior in an increasingly computer-mediated world. *Child abuse & neglect*, 44.

Bogdanova, D., Rosso, P., and Solorio, T. (2012). On the impact of sentiment and emotion based features in detecting online sexual predators. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, WASSA '12.

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y., Strope, B., and Kurzweil, R. (2018a). Universal sentence encoder. *CoRR*, abs/1803.11175.

Cer, D., Yang, Y., yi Kong, S., Hua, N., Limtiaco, N., St. John, R., Constant, N., Guajardo-Céspedes, M., Yuan, S., Tar, C., Sung, Y. H., Strope, B., and Kurzweil, R. (2018b). Universal sentence encoder for English. *EMNLP 2018 - Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Proceedings*, pages 169–174.

Cheong, Y., Jensen, A. K., Guonadottir, E. R., Bae, B., and Togelius, J. (2015). Detecting predatory behavior in game chats. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(3), Sep.

Chiu, M. M., Seigfried-Spellar, K. C., and Ringenberg, T. R. (2018). Exploring detection of contact vs. fantasy online sexual offenders in chats with minors: Statistical discourse analysis of self-disclosure and emotion words. *Child abuse & neglect*, 81.

Cocospy. (2019). Cocospy. https://www.cocospy.com.

Easemon Inc. (2019). iKeyMonitor. https://ikeymonitor.com.

Eriksson, G. and Karlgren, J. (2012). Features for modelling characteristics of conversations: Notebook for pan at clef 2012. In *CLEF (Online Working Notes/Labs/Workshop)*.

Escalante, H. J., Villatoro-Tello, E., Juárez, A., Montes-y Gómez, M., and Villaseñor, L. (2013). Sexual predator detection in chats with chained classifiers. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, June.

FlexiSPY. (2019). FlexiSPY. https://www.flexispy.com.

Inches, G. and Crestani, F. (2012a). Overview of the International Sexual Predator Identification Competitin at PAN-2012. *Working Notes Papers of the CLEF 2012 Evaluation Labs*, (May).

Inches, G. and Crestani, F. (2012b). Overview of the international sexual predator identification competition at pan-2012. In *CLEF (Online Working Notes/Labs/Workshop)*.

Inches, G. and Crestani, F. (2012c). Overview of the international sexual predator identification competition at pan-2012. In *CLEF*.

Kontostathis, A., Garron, A., Reynolds, K., West, W., and Edwards, L. (2012). Identifying predators using chatcoder 2.0. In *CLEF (Online Working Notes/Labs/Workshop)*.

Madigan, S., Villani, V., Azzopardi, C., Laut, D., Smith, T., Temple, J. R., Browne, D., and Dimitropoulos, G. (2018). The prevalence of unwanted online sexual exposure and solicitation among youth: A meta-analysis. *Journal of Adolescent Health*, 63(2):133 – 141.

McGhee, I., Bayzick, J., Kontostathis, A., Edwards, L., McBride, A., and Jakubowski, E. (2011a). Learning to identify internet sexual predation. *International Journal of Electronic Commerce*, 15(3):103–122.

McGhee, I., Bayzick, J., Kontostathis, A., Edwards, L., McBride, A., and Jakubowski, E. (2011b). Learning to identify Internet sexual predation. *International Journal of Electronic Commerce*, 15(3):103–122.

Miah, M. W. R., Yearwood, J., and Kulkarni, S. (2011). Detection of child exploiting chats from a mixed chat dataset as a text classification task. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, December.

Mikolov, T., Yih, W. T., and Zweig, G. (2013). Linguistic regularities in continuous spaceword representations. *NAACL HLT 2013 - 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Main Conference*, pages 746–751.

Morris, C. and Hirst, G. (2012). Identifying sexual predators by svm classification with lexical and behavioral features. In *CLEF (Online Working Notes/Labs/Workshop)*.

Olson, L. N., Daggs, J. L., Ellevold, B. L., and Rogers, T. K. K. (2007). Entrapping the innocent: Toward a theory of child sexual predators' luring communication. *Communication Theory*, 17(3).

O'Connell, R. L. (2003). A typology of child cybersexploitation and online grooming practices. Preston: University of Central Lancashire, Cybersex Research Unit.

Parapar, J., Losada, D. E., and Barreiro, A. (2012). A learning-based approach for the identification of sexual predators in chat logs. In *CLEF (Online Working Notes/Labs/Workshop)*.

Peersman, C., Vaassen, F., Van Asch, V., and Daelemans, W. (2012). Conversation level constraints on pedophile detection in chat rooms. In *CLEF (Online Working Notes/Labs/Workshop)*.

Pendar, N. (2007). Toward spotting the pedophile telling victim from predator in text chats. In *International Conference on Semantic Computing (ICSC 2007)*, Sep.

Pennebaker, J. W., Chung, C. K., Ireland, M. E., Gonzales, A. L., and Booth, R. J. (2011). The development and psychometric properties of liwc2007.

Perverted Justice Foundation. (2019). Perverted Justice. `www.perverted-justice.com`. Accessed: 2019-11-08.

Vartapetiance, A. and Gillam, L. (2014). "our little secret": pinpointing potential predators. *Security Informatics*, 3(1):3, Sep.

Villatoro-Tello, E., Juárez-González, A., Escalante, H. J., y Gómez, M. M., and Pineda, L. V. (2012). A two-step approach for effective detection of misbehaving users in chats. In *CLEF (Online Working Notes/Labs/Workshop)*.