Toward Spotting the Pedophile
Telling victim from predator in text chats

Nick Pendar
Human Computer Interaction Program
Iowa State University
U.S.A. 50011
pendar@iastate.edu

Abstract

This paper presents the results of a pilot study on using automatic text categorization techniques in identifying on-line sexual predators. We report on our SVM and k-NN models. Our distance weighted k-NN classifier reaches an f-measure of 0.943 on test data distinguishing the child and the victim sides of text chats between sexual predators and volunteers posing as underage victims.

1 Introduction

There is no doubt that the Internet has forever changed how almost everyone lives, works, and plays; and that includes criminals. Of particular concern are on-line pedophiles who "groom" children, that is, who meet underage victims on-line, engage in sexually explicit text or video chat with them, and eventually convince the children to meet them in person. According to the National Center for Missing & Exploited Children [11] and the Office of Juvenile Justice and Delinquency Prevention, one out of every seven children receives an unwanted sexual solicitation on-line. Harms [4] operationally defines grooming as

> a communication process by which a perpetrator applies affinity seeking strategies, while simultaneously acquiring information about and sexually desensitizing targeted victims in order to develop relationships that result in need fulfillment.

Need fulfillment, in this case, is defined as "physical sexual molestation." The customary way to catch these sexual predators is for trained law enforcement officers or volunteers to pose as children in on-line chat rooms; however, on-line sexual predators always outnumber the law enforcement officers and volunteers.

There is now a great need for software applications that can flag suspicious on-line chats automatically, either as a tool to aid law enforcement officials or as parental control features offered by chat service providers. Given the recent advances in text categorization, such an application is definitely within reach. A group of researchers at Iowa State University have initiated a research program entitled Study for the Termination of Online Predators (S.T.O.P.),[1] which focuses on the communicative, computational and criminological aspects of the grooming process. This paper presents a pilot study toward building an automatic recognition system of on-line predators. In addition to reporting our findings, one objective of this paper is to increase awareness in the research community of this important issue and the attainability of a solution.

2 Data

One major problem in this enterprise is the data acquisition bottleneck. There can generally be two types of on-line text chat with sexually explicit content that are of interest to this project: (I) interaction between a sexual predator and what that individual believes to be a victim, and (II) consensual interaction between two adults. The first type itself will have its subtypes. The predator could be interacting with a real underage victim, or he could unknowingly be interacting with a law enforcement officer or a volunteer. These types of interactions are summarized below:

I Predator/Other

- (a) Predator/Victim (victim is underage)
- (b) Predator/Pseudo-Victim (volunteer posing as child)
- (c) Predator/Pseudo-Victim (law enforcement officer posing as child)

[1] www.stopandhelp.org

II Adult/adult (consensual relationship)

Ideally, in order to build a system to flag an interaction as suspicious, i.e., of type (Ia), we at least need to have access to representative samples of interactions of type (II) as well as (Ia). However, chat service providers do not customarily archive adult instant messaging chats, and even if they did, they would not make those archives available to the public. Accumulating such data requires the informed consent of the participants. In addition, access to archives of type (Ia) instant messaging text chats is also difficult. Getting access to (Ic) type of data is not without problems either. Significant privacy and legal issues need to be resolved. Therefore, even a simple feasibility study for this type of project immediately faces major data acquisition problems none of which is necessarily technical.

Fortunately, the next best thing, i.e., samples of type (Ib) interactions, are available on-line. The Web site `www.perverted-justice.com`, which is run by a group of volunteers, aims at making it difficult for pedophiles to meet underage victims on-line. This Web site "recruits volunteer contributors who pose as underage children [typically ages 10–15] in chatrooms."[2] When a pedophile has been found, the Web site posts archives of all text chats with them on-line. We decided to use these data in order to gage the viability of a system for automatic recognition of on-line sexual predators.

For this study, we did not have access to any negative data, i.e., samples of type (II) interactions; therefore, we decided to see if given a type (Ib) text log, we can automatically distinguish between the pseudo-victim and the predator, with the assumption that a positive result would support the hypothesis that it is possible to automatically flag suspicious on-line chats. Normally, the two sides of a conversation are on the same topic at any given point of their interaction, which means adjacent chat lines are more or less on the same topic. If this is true, then one might argue that a simple text categorization technique should not be able to distinguish between the two sides of the conversation. But while the two sides of the conversation may be on the same topic, they may be using very different subsets of the language of conversation. A (pseudo-)child will presumably use a very different type of language than a pedophile grooming the (pseudo-)child. If we can model these differences with text categorization software, it should also be possible to extend the model to distinguish other types of interlocutors. A suspicious interaction can then be defined as any interaction between a likely child and a likely predator.

We downloaded archives of text chats between sexual predators and their pseudo-victims available on `www.perverted-justice.com`. At the time of the study, we had a collection of 701 text logs, i.e., conversation between 701 sexual predators and what they thought were underage victims, *the pseudo-victims*. The lengths of the text logs ranged between 269 and 42,220 words including the screen names and timestamps. The median length of text logs was 2,629 words. Longer text logs included conversations taking place in more than one sitting. The whole collection contains 2,603,681 words including the screen names and timestamps. Henceforth, we shall call this dataset the *PJ corpus*.

Each file was then divided into two: one including the chat lines produced by the predator and one including the chat lines produced by the pseudo-victim. This means that after the division we had a corpus with 1,402 files with 701 files containing the predators' lines and 701 files containing the corresponding pseudo-victims' lines. This corpus was then divided up into a training set (1,122 files) and a test set (280 files). The sampling preserved the proportion of files containing the pseudo-victims' input and those of the predators (i.e., 140 predators' files and 140 pseudo-victims' files). The test set was not used in any aspect of feature selection or learning. It was strictly used to test the generlizability of the models built based on the data from the training set.

Viewers of `www.perverted-justice.com` rate each predator on a five-point scale based on how "slimy" they perceive each predator to be, 1 being the lowest and 5 the highest levels of "sliminess." The distribution of the sliminess measures of the files is presented in Figure 1. As can be seen, the distribution is highly right-skewed. The median rating is 4.535. The number of votes per file is also very spread out, ranging from 13 to 20,000 votes, with a mean of 564.8. The reason for this is that the site posts direct links to the text logs of its top five "slimiest" catches; hence the greatest number of votes for the individuals with highest "sliminess" ranking. This raises the question of whether the number of votes influences the overall ranking. An independent samples t-test between the files with higher ratings (above median) and those with lower ratings (below median) revealed that there is no significant difference between the number of votes for files with higher ratings and those with lower ratings. However, in order to make sure that our test data contained a representative sample of all ratings, during the selection of the test data, we also made sure that the predator files included an equal number of those with low (below median) and high (above median) "sliminess" ratings.

3 The Classifiers

The PJ corpus was used in a feasibility study to assess whether text categorization techniques can be used in identifying on-line child-predator communication. We trained

[2] `www.perverted-justice.com/guide/`

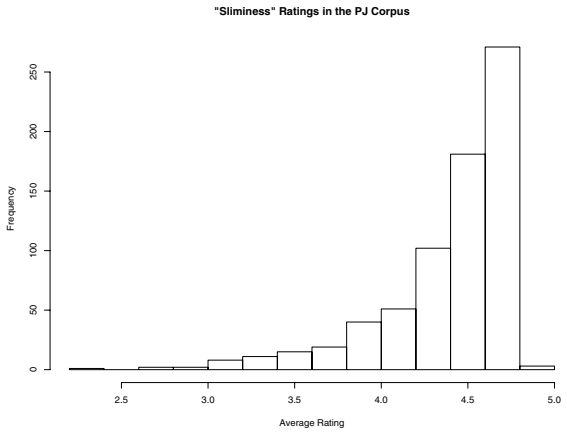

Figure 1. Distribution of "sliminess" measures in the PJ Corpus

a series of SVM and distance-weighted k-NNclassifiers. k-NN classifiers are discussed in detail in chapter 8 of [8], SVMs are discussed in [1], [3], and [5]. The results of some of these classifiers we are reporting in this paper.

k-NN classification is a form of memory based learning [8] in which each document d is represented as an n-dimensional vector of weights, $\vec{d} = \langle a_1(d), a_2(d), \ldots, a_n(d) \rangle$. Each weight $a_i(d)$ is a measurement of a feature in text, typically a term, measured as normalized tf-idf (see below). In a simple k-NN classifier, a new document x_q is assigned the category that is most common among the k-nearest (most similar) documents of the training set D in the $\mathcal{R}^n$ feature space. An array of similarity measures can then be used to find the nearest neighbors. We used the Euclidean distance calculated using the formula presented in (1) to find the nearest neighbors to the test documents.

$$\delta(x_q, d_i) = \sqrt{\sum_{j=1}^{n} (a_j(x_q) - a_j(d_i))^2} \tag{1}$$

In a distance-weighted k-NN classifier, documents that are closer to a query document x_q are given more weight than those farther away among the k-nearest neighbors of x_q. Therefore, the category of x_q will be calculated as follows:

$$\hat{f}(x_q) = \underset{c \in C}{\operatorname{argmax}} \sum_{i=1}^{k} w_i \delta(c, f(d_i)) \tag{2}$$

where $\hat{f}(x_q)$ is the category assigned to the query document x_q; C is the set of possible categories; $f(d_i)$ is the category of the document $d_i \in D$; $\delta(c, f(d_i)) = 1$ iff $c = f(d_i)$ and $\delta(c, f(d_i)) = 0$ otherwise; and w_i is defined as follows:

$$w_i \stackrel{\text{def}}{=} \frac{1}{\delta(x_q, d_i)^2} \tag{3}$$

If $\delta(x_q, d_i) = 0$, then $\hat{f}(x_q) = f(d_i)$, and if there are more than one such cases, then $\hat{f}(q)$ will be assigned by a majority vote.

Support vector machines (SVMs) are used for binary classification. In SVM learning a hyperplane is found between input vectors with maximum distance between the two groups in the training data, i.e., we find a hypothesis that guarantees the lowest true error rate.

4 Feature Extraction

We tried using unigrams, bigrams and trigrams from the training data as features. In text categorization and information retrieval, function words are usually filtered out in preprocessing using a list of most common such words known as a stop list. On-line chats have their own vocabulary and spelling rules, which renders any standard stop list useless. We built a stop list of the 79 most frequent word types in the corpus. A longer word list would include content words. The non-standard language of the text chats and frequent misspellings also provide a challenge for stemming. In this pilot study, we did not try stemming the words nor did we try to correct spelling. Stemming text chat words and spelling correction merit their own research projects and we shall not address them here. Another problem that on-line text chats present is the repetition of the same letter to imitate lengthening of words in speech for emphasis. For example, the word *no* can be spelled as *no*, *noo*, *nooo*, etc. In preprocessing in addition to removing screen names, time stamps and punctuation marks, we circumvented the problem of repeated letters by simply replacing any repeated letters in words by a single instance of that letter. This introduces new misspellings in text, but results in more consistent spelling. In the future, this process can be integrated with a spell checker so that legitimate repetitions of letters in words are not altered. Removing punctuation marks may have the disadvantage of losing some potential information conveyed through the use of emoticons. Emoticons combinations of punctuation marks and possibly letters conveying emotions, e.g., `:-)` or `;-)`. In the future, we will explore the usefulness of these as well.

After removing all the stop words, we extracted the n-grams by making three passes over each text chat line using a window of one, two, and three words, respectively. The n-grams and their frequencies for each file for each chat participant (i.e., pseudo-victim or predator) were recorded. The word types in bigrams and trigrams are ordered alphabetically in order to (i) capture the semantic similarity

of stretches of text containing the same words in different orders, (ii) minimize interdependence among features, and (iii) limit the number of features. In total, 295,356 unigrams, 360,284 bigrams, and 24,440 trigrams were extracted.

We performed dimensionality reduction by feature extraction.[3] Several feature extraction functions (FEF) are typically used in the text categorization literature. Sebastiani [10] lists some of the most popular FEFs. We used a combination of document frequency (df) threshold and odds ratios of terms for feature selection. The choice of odds ratio as a FEF was based on the fact that odds ratio has previously been shown to be a better FEF than Chi-square, information gain, cross entropy and mutual information. See [2] and [9], for example; also see [12] and [6] for other comparative studies of several feature extraction methods. Admittedly, the superiority of odds ratio statistics may not necessarily carry over to this new application of text categorization with these new data. We shall leave experimenting with other FEFs to future research.

In automatic text categorization, it is also customary to eliminate digits in feature extraction. We decided not to do that since numbers could potentially provide important information as to the age of the parties involved or other information about them such as their dress size etc. that can be used as clues in identifying suspicious interactions.

During feature extraction, we ignored n-grams that appeared only in one document (i.e., $df(t_j) = 1$) and those that appeared in more than 95% of the documents in the training set (i.e., $df(t_j) > 0.95 \times |D|$). This led to 10,766 unigrams, 43,383 bigrams, and 13,098 trigrams. We then calculated the odds ratios of each of these n-grams with respect to the categories $C = \{\text{pseudo-victim}, \text{predator}\}$. The odds ratio of the n-gram t_j occurring in the category c_i was calculated as follows:

$$OR_{c_i}(t_j) = \frac{p(t_j|c_i)(1 - p(t_j|\bar{c}_i))}{(1 - p(t_j|c_i))p(t_j|\bar{c}_i)} \tag{4}$$

where $OR_{c_i}(t_j)$ is the odds ratio of the term (n-gram) t_j with respect to the category c_i, and $\bar{c}_i$ is the complement of c_i. The odds ratios were then averaged using the following formula:

$$OR_{avg}(t_j) = \sum_{i=1}^{|C|} OR_{c_i}(t_j)p(c_i) \tag{5}$$

The above probabilities are defined over the space of the documents in the training data, and in this case since there were only two categories, and there was an equal number of texts in each category, then $p(c_i) = 0.5$.

[3]Note that in this paper we are using *term*, *n-gram* and *feature* interchangeably.

When the average odds ratios were calculated for all the n-grams in the training set, we built nine feature sets by extracting 5,000, 7,500 and 10,000 unigrams, bigrams and trigrams with the highest average odds ratios.

Then for each document in the training and test sets, we calculated the tf-idfs of the extracted features. This gave us nine different representations for each document depending on whether the features were unigrams, bigrams or trigrams and whether the dimensionality of each vector representation was 5,000, 7,500 or 10,000. The tf-idfs were calculated according to the following formula:

$$tf\_idf_d(t_j) = tf_d(t_j)log(\frac{|D|}{df(t_j)}) \tag{6}$$

where $tf\_idf_d(t_j)$, $tf_d(t_j)$, and $df(t_j)$ are, respectively, the td-idf, term frequency, and document frequency of the term (n-gram) t_j with respect to document d; and D is the set of documents in the training set. The tf-idfs then get normalized by multiplying them by the constant

$$\frac{1}{\sqrt{\sum_{j=1}^{n} tf\_idf_{t_j}^2}}$$

where $\sum tf\_idf_{t_j}^2$ is the sum of the squares of the tf-idfs of all the selected features.

5 Experiments and Results

We tested the effectiveness of the nine representations in a SVM and a distance-weighted k-NN classifier.

To evaulate the performance of the models, we calculated the micro- and macro-averages of precision and recall. See [10] for a discussion of these measurements. The macro- and micro-averages were generally very close to each other throughout. In this paper, the micro-average f-measures are reported. F-measures are calculated as follows: $2\pi\rho/(\pi + \rho)$, where π and ρ are precision and recall, respectively. Micro- and macro-average precision and recall are calculated using the formulas in (7)–(10) where $\hat{\pi}^{\mu}$ is the micro-average precision, $\hat{\rho}^{\mu}$ is the micro-average recall, $\hat{\pi}^{M}$ is the macro-average precision, and $\hat{\rho}^{M}$ is the macro-average recall.

$$\hat{\pi}^{\mu} = \frac{TP}{TP + FP} \tag{7}$$

$$\hat{\rho}^{\mu} = \frac{TP}{TP + FN} \tag{8}$$

$$\hat{\pi}^{M} = \frac{\sum_{i=1}^{|C|} \hat{\pi}_i}{|C|} \tag{9}$$

$$\hat{\rho}^{M} = \frac{\sum_{i=1}^{|C|} \hat{\rho}_i}{|C|} \tag{10}$$

Terms	F-Measure		
	5000	7500	10000
Unigram	0.558	0.415	0.415
Bigram	0.575	0.545	0.545
Trigram	0.893	0.908	0.908

Table 1. F-measures of the SVM models with different dimensionalities.

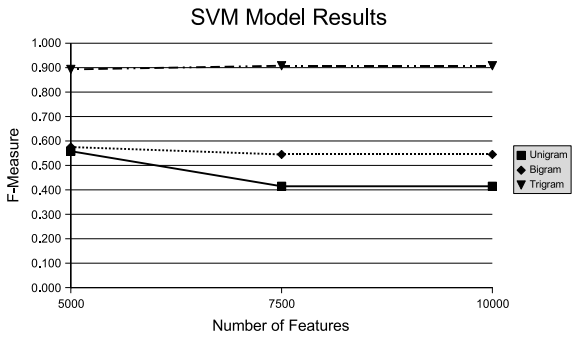

Figure 2. Unigram model results

Terms	k Value	F-Measure		
		5000	7500	10000
Unigram	5	0.546	0.586	0.814
	10	0.675	0.571	0.854
	15	0.607	0.575	0.818
	20	0.586	0.561	0.811
	25	0.579	0.582	0.818
	30	0.571	0.575	0.807
Bigram	5	0.500	0.500	0.500
	10	0.500	0.511	0.500
	15	0.582	0.500	0.500
	20	0.514	0.500	0.500
	25	0.504	0.500	0.675
	30	0.500	0.500	0.779
Trigram	5	0.504	0.500	0.514
	10	0.504	0.500	0.871
	15	0.500	0.532	0.925
	20	0.504	0.507	0.918
	25	0.529	0.500	0.936
	30	0.511	0.500	**0.943**

Table 2. F-measures of the k-NN classifier with different k values and dimensionalities.

We used the machine learning environment YALE [7] to build a series of SVM models for our data with a linear kernel. We tried unigrams, bigrams and trigrams as features with a dimensionality $n \in \{5000, 7500, 10000\}$. The trigram models included only trigrams as features; bigrams and unigrams were not included in the model in order to prevent feature interdependence. Similarly, the bigram models included only bigrams and no unigrams. The results of these models are presented in Table 1 and Figure 2. As can be seen, the best results with the SVM models were achieved with the trigram models with $n \in \{7500, 10000\}$. The worst results (worse than chance) are with the unigram models.

With the k-NN classifier, we tried $k \in \{5, 10, 15, 20, 25, 30\}$ on the document representations using unigrams, bigrams, or trigrams with a dimensionality $n \in \{5000, 7500, 10000\}$. The best result was achieved using trigram features and the dimensionality of 10,000 and $k = 30$. Table 2 and Figures 3–5 present a summary of the results of this study. It is interesting to see that bigram features generally yield the worst results here. Only do the classifiers using 10,000 bigram features and $k \in \{25, 30\}$ perform significantly better than chance. The unigram and trigram models, on the other hand, generally perform better than the bigram models. This suggests that what distinguishes the two sides of these conversations may not simply be the words they use, but their phrases, i.e., the manner in which they put their words together. Overall, 10,000 features were needed for satisfactory performance, the best of which is 0.943 f-measure for trigrams with $k = 30$. We will, of course, be able to perform more aggressive dimensionality reduction by cleaning out the spelling more and stemming as well as experimenting with more sophisticated dimensionality reduction techniques.

We also tried the same experiments on only the predator part of the PJ corpus to see to what extent it is possible to distinguish between predators with high and low "sliminess" measures, i.e., above and below median. Our k-NN classifier hardly performed better than chance at all parameter settings. This supports our intuition that the predator and the child side of the text chats in PJ corpus in fact do use different subsets of the English language while the predators themselves use a similar type of language. In addition, our experiments show that these differences are best captured by the trigrams.

The difference in "sliminess" of the predators, then, may not lie in the type of language they use but in the communication strategies that they employ. It is known that the grooming process has at least the following three stages [4]: *affinity seeking* (building trust), *information acquisition* (determining if a likely victim has been found), and *sexual desensitization* (exposing victim to sexually explicit language and/or images). It may well be that "sliminess" lies in the length and intensity of each of these stages throughout the grooming process. Further studies in this regard are required if we want to be able to rank likely predators based

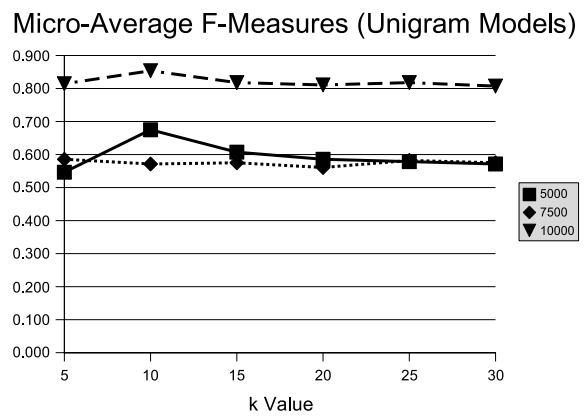

Figure 3. Unigram model results

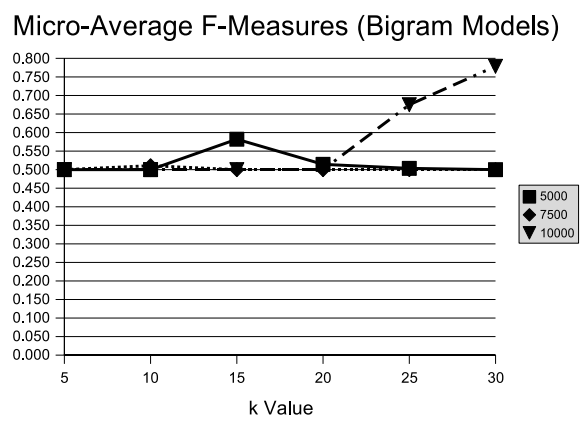

Figure 4. Bigram model results

on the risk they present.

A simple error analysis of the best-performing model (the k-NN model with $k = 30$ and $n = 10000$) revealed that 6.4% of the pseudo-victims were misclassified as predators. Among the predators, 7.1% of the ones with lower than median "sliminess" measures were misclassified as victims. The observed error rate for the predators with higher than median "sliminess" measures, on the other hand, was only 2.9%. In other words, the less "slimy" predators were harder to recognize, which is not surprising.

6 Discussion

This study showed that given a file containing the contents of a text chat between a sexual predator and a pseudo-victim, it is possible to automatically distinguish between the two sides with very high accuracy. One question that remains is whether these results carry over to data from real

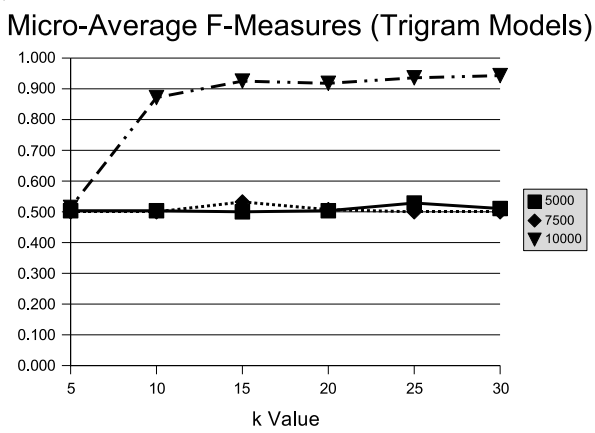

Figure 5. Trigram model results

victims. At this stage, we are in the process of acquiring such data. Another remaining challenge in this project is distinguishing type II communication from our target type Ia communication. To do this we will need to be able to identify underage interlocutors among adult ones. This study has shown that this task is definitely possible.

One crucial issue that needs to be addressed in future research is the fact that text chats develop over time, and we would like to be able to flag suspicious interactions as early as we can without having to wait for a complete script. This means that we need to be able to reliably gage the risk of the conversation after each line of chat has been entered. Here it may be useful to identify the stages of communication, as in *affinity seeking*, *information acquisition*, or *sexual desensitization*. Therefore, research is also needed for automatic identification of these communication strategies.

Acknowledgements

I am indebted to Chad Harms for introducing this line of research to me, and to Brian Monahan for his comments on this paper. I also thank the anonymous reviewes of this paper for their insightful comments.

References

[1] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[2] M. F. Caropreso, S. Matwin, and F. Sebastiani. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In A. G. Chin, editor, *Text Databases and Document Management: Theory and Practice*, pages 78–102. Idea Group Publishing, Hershey, US, 2001.

[3] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[4] C. Harms. Grooming: An operational definition and coding scheme. *Sex Offender Law Report*, forthcoming.

[5] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142. Springer, 1998.

[6] S. L. Lam and D. L. Lee. Feature reduction for neural network based text categorization. In A. L. Chen and F. H. Lochovsky, editors, *Proceedings of DASFAA-99, 6th IEEE International Conference on Database Advanced Systems for Advanced Application*, pages 195–202, Hsinchu, TW, 1999. IEEE Computer Society Press, Los Alamitos, US.

[7] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler. YALE: Rapid prototyping for complex data mining tasks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006)*, 2006.

[8] T. Mitchell. *Machine Learning*. Computer Science Series. McGraw Hill, New York, 1997.

[9] D. Mladenic. Feature subset selection in text-learning. In *European Conference on Machine Learning*, pages 95–100, 1998.

[10] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[11] J. Wolak, K. Mitchell, and D. Finkelhor. Online victimization of youth: Five years later. National Center for Missing & Exploited Children Builletin 07-06-025, National Center for Missing & Exploited Children, Alexandia, VA, 2006.

[12] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In D. H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 412–420, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.