Title: Learning to Identify Internet Sexual Predation

Authors: India McGhee, Jennifer Bayzick, April Kontostathis*
Mathematics and Computer Science, Ursinus College

Lynne Edwards, Alexandra McBride, Emma Jakubowski
Media and Communication Studies, Ursinus College

* April Kontostathis will be the corresponding author for the submission. Please direct correspondence to:

April Kontostathis
Math and CS Dept
Ursinus College
PO Box 1000, 601 Main St.
Collegeville PA 19426
akontostathis@ursinus.edu
610-409-3650 (voice)
610-409-3660 (fax)

Learning to Identify Internet Sexual Predation

Abstract:

This paper describes progress on an ongoing project that integrates both communication theories and computer science algorithms to create a program that can detect the occurrence of sexual predation in an online social setting. Though there has been much work exploring social media in general, this particular aspect of online social interaction remains largely untapped.

In particular, our study attempts to evaluate and categorize the strategies used by online sexual predators in their attempts to develop relationships with children using the Internet. Our software system, ChatCoder, is designed to decide which lines in the chat log contain predatory language. The labels we use are based on the communicative model for online predation that we have developed and are refining. This article describes the communicative model in detail, with specific descriptions of the choices we made as we refined the model.

In previous work we developed phrase-matching and rule-based approaches to classify and label lines of chat logs. In the current work, we expand these techniques and use machine learning algorithms to classify posts. Our machine learning system leveraged the phrase-matching and rule-based systems to identify appropriate attributes for our supervised learning algorithms. A traditional bag-of-words approach was insufficient because posts are typically very short and contain many variations of words (due to misspellings and netspeak usage).

Our machine learning experiments confirmed that the rules we developed are adequate to identify the coding rules. Neither decision trees nor instance-based learning algorithms were able to significantly improve upon the 68% accuracy we were able to achieve using the rule-based methods employed by ChatCoder 2.

The contributions of this paper are as follows:

- Development of techniques for matching human hand-coding of transcripts. This result can be leveraged to other applications thereby reducing the amount of time researchers in communications theory must spend training human coders. Online language (or computer-mediated communications, CMC, as it is termed by communications researchers) is constantly evolving, and human coders must occasionally be retrained even when working on the same project. Automated methods of coding, even if they require training sets based on human coding of a subset of the documents, would reduce the amount of time spent hand-coding large volumes of transcripts.
- Application of machine learning approaches to text classification at the post level. We successfully used both decision trees and nearest-neighbor classifiers for text classification.

- Evidence that these techniques are viable and can identify potential attributes that may be useful in other applications.

1. Introduction

The National Center for Missing and Exploited Children (NCMEC) reported in a 2008 survey that approximately 1 in 7 youth (ages 10-17) are approached or solicited for sexual purposes through the Internet. The NCMEC established a CyberTipLine for reporting cases of child sexual exploitation, and the magnitude of calls received is staggering, with 47,670 reports of "Online Enticement of Children for Sexual Acts" from March 1998 to January 2010 [8]. By comparison, there were approximately 40,000 reports as of October 2008.

We are continuing a study that attempts to evaluate and categorize the strategies used by online sexual predators in their attempts to develop relationships with children using the Internet. Working with a program called ChatCoder, we redesigned the method employed by the program to decide which lines in the chat log contain predatory language.

Communications researchers define two primary goals for content analysis: to describe communication, and to draw inferences about its meaning [12]. In this paper we describe a rule-based approach for identifying predatory communication within a chat log and compare our algorithmic effort to the results obtained by various machine learning algorithms. The truth set we are using for our work was produced by trained communications analysts. The chat logs we have been working with come from the Perverted Justice (PJ) website [11] and are discussions between a sexual perpetrator, who has been convicted of soliciting a minor over the Internet for sexual activity, and a trained PJ volunteer posing as an adolescent.

Like most of the commercial products we have seen, ChatCoder 1 was based on a simple keyword matching technique [4]. This approach has many shortcomings. We have found that the rule-based technique used in ChatCoder 2 provided an overall improvement in intercoder reliability by a maximum of 13.21% and an average of 5.81% on a small set of chat transcripts using 9 classes [6]. Intercoder reliability is a well-known metric used by researchers in communication studies for determining the consistency of labeling between two coders or coding systems.

In the sections that follow we explain how we expanded the number of transcripts used for testing and reduced the number of classes. An explanation of our revised model appears along with a description of related work in Section 2, and an overview of our dataset and attributes appears in Section 3. A detailed description of the experimental design and our results appears in Section 4. We offer our conclusions in Section 5.

A number of social networking sites involve chats or forum discussions, and as these sites draw younger users, they also draw cyber criminals. Our analysis of chat data can provide interesting insights and may inform the development of new tools for site monitoring.

2. Related Work and Communicative Model Development

This project integrates both communication theories and computer science algorithms. Our long-term goal is to create a program that can detect an occurrence of predation in an online social setting. Though there has been much work exploring social media, this aspect of online social interaction remains largely untapped.

2.1 Related Work

Although there have been a number of research projects involving the parsing of chat logs, there have been few that do so with predation in mind. To our knowledge, there are three such projects besides our own. Pendar has had some success analyzing chat transcripts to differentiate between the victim and the predator [10]. The study by Hughes, et al. focuses on the distribution of child pornography through peer-to-peer networks [3]. Recently, Adams and Martel conducted research to create a program that can detect and extract the topic of discussion, and Yin, et al. used supervised methods to differentiate between posts containing harassment and those that contained innocent discussion in a chat room or message board environment [1, 15]. There are also commercial computer programs that attempt to police chat conversations, but they are generally lacking in true analysis capabilities as none of them are based on communication theory [5].

2.2 Communicative Model Development

Olson, et al. established a luring communication theoretical model (LCT) that defines five phases of predation: gaining access, deceptive trust development, grooming, isolation, and approach [9]. This model was expanded for online predation and operationalized by Leatherman [7]. The Leatherman model contained nine classes, and in earlier work we developed rules for identifying labeling posts using these nine categories [6].

2.2.1 Olson and Leatherman Models

Gaining access, as defined by Olson, et al. represents the first step in the luring process wherein the predator must be "motivated and able to gain access to potential victims and their families." Primarily, then, gaining access involves exchanging personal traits of both the predator and the victim, as well as the strategic placement of the predator. In online predation, a predator gains access to a minor through media such as instant messaging forums, chat rooms, and social networking sites such as MySpace and Facebook. For instance, a predator would place himself in a chat room frequented by minors.

Olson, et al. define deceptive trust development as "a perpetrator's ability to cultivate relationships with potential victims and possibly their families that are intended to benefit the perpetrator's own sexual interest." In the arena of online predation, this is divided into four sub-categories: personal information, relationship details, activities, and compliments. The exchange of personal information involves details about the victim's and predator's actual locations, ages, names, computer locations (i.e. in the bedroom, basement, etc.), birthdays, cell or home phone numbers, and pictures of themselves. Relationship information includes discussion of feelings and attitudes toward maintaining, building, and dismantling their relationships with each other, friends, significant others, and family members. Activities, a broad category, is defined primarily as preferred social behaviors shared by both the predator and victim, including but not limited to music, movies, books, sports, hobbies, and favorite foods. Compliments involve the predator or victim offering praise for one another's appearance, activities, and personal information with the intention of making the victim view the predator in a positive, appreciative light.

Deceptive trust occurs and is affirmed throughout the entire communication between the predator and the victim. Indeed, incurring the trust of the victim is essential to the success of the later stages of the entrapment cycle: isolation and approach. When the victim trusts the predator, the offender begins to groom the minor to accept offers of sexual contact. Grooming is “the subtle communication strategies that sexual abusers use to prepare their potential victims to accept the sexual conduct.” [9] Thus communication that functions as grooming does not directly lead to sexual contact, but instead desensitizes the victim to sexual remarks or foul language. Successful grooming leaves the victim unaware that any process is underway. There are two sub-categories of grooming - communicative desensitization and reframing.

Communicative desensitization refers to the offender purposefully and frequently using vulgar sexual language in an attempt to desensitize the victim to its use. Additionally, the perpetrator will often attempt to encourage the minor's interest in sexual subjects with the goal of perpetrating future abuse. In terms of online predation, this can be achieved by sending pornographic images and using sexual slang terms or netspeak in lieu of every day words (i.e. “welcum” instead of “welcome”).

Reframing occurs when sex offenders endeavor to make the victim comfortable with experiencing sexual advances over the Internet. From Olson, et al., reframing is “contact or sex play between victim and predator that may be communicated in ways that would make it beneficial to the victim later in life.” To this end, sexual conversation is presented in a positive light and is often referred to as a learning experience, a game to be played, or an important skill to learn in order to participate in loving relationships in the future.

Beyond grooming, physically and emotionally isolating the victim is essential to the sexual predator, be it online or in the real world. Physical isolation is defined as arranging to spend time alone with the victim, and mental isolation is increasing emotional dependency upon the predator for things like friendship and guidance. While complete physical isolation cannot occur over the Internet, the predator achieves isolation by making sure the victim chats without supervision. Predation is most successful with minors who are isolated from support networks, be it by low paternal or maternal relationships or by having very few friends. This information is gleaned through online communications by asking questions about the minor's social life, by providing sympathy and support in reaction to their situation, and by questioning the strict rules of the parent. The predator seeks to isolate the victim and then to fill the social gaps in the victim's life as a tool to facilitate abuse and gain control of the victim.

When the predator has established the victim's trust, commenced grooming, and isolated the minor from support networks, the predator attempts to approach the victim by suggesting that they meet for sexual purposes. In the LCT model, Olson, et al., define approach as “the initial physical contact or verbal lead-ins that occur prior to the actual sexual act.” In the online model of luring communication, approach is the final step when the predator requests to meet the victim offline with the intent of beginning a sexual relationship.

2.2.2 Updated model

We recently revised the model as it pertains to online predation. The Olson and Leatherman models were deemed too complex for the short communication bursts that take place within the context of a chat conversation, and the word “luring,” as used by Olson and Leatherman, does not seem to apply. Minors often enter chat rooms and engage in conversations with strangers; therefore, gaining access is trivial. Furthermore the two key aspects that make the encounter predatory are the ages of the victim and predator, and the approach (the attempt to actually meet). Therefore, we condensed and simplified the model into four classes:

200 – Exchange of personal information
600 – Grooming
900 – Approach
000 – Lines containing none of the classes

The exchange of personal information (class 200) often includes questions about age, gender and location. Topics such as number of friends, previous or current boyfriends, and likes/dislikes are also discussed. The predators tend to use this information to indicate that they have something in common with the potential victims and/or to gauge their support structures (parents divorced or recent moves).

Grooming (class 600) involves the use of sexual terminology, whether or not it is in context (e.g. discussing the virginity status of the victim, or just using “cum” in place of the “come” in a line that would otherwise be more innocent). Topics that might be considered reframing would appear here also (e.g. “I can teach you to do that” used during a discussion about the sexual experience of the victim).

Approach (class 900) includes trying to obtain the victim’s phone number or address, arrange a meeting, or keep the relationship between the victim and predator a secret from parents or authorities.

Finally some lines are not coded at all (class 000), because they simply keep the conversation going (e.g. yeah, lol) or appear to be truly innocent (e.g. moves in a game).

Though the representation of the four categories as numerical values leads us to the implication that they are treated as ordinal, this is not the case. ChatCoder 2 treats the categories as nominal values, meaning that if a category is coded incorrectly, there is no varying degree of correctness – coding personal information as grooming is equally incorrect as coding the same line as approach.

3. Dataset and Attributes

When beginning this project, chat transcripts in text files were downloaded from the Perverted Justice website (PJ) [11]. The files contain entire conversations between adults posing as young teens and convicted sexual offenders. As of April 2010, the PJ site had 506 transcripts. We randomly selected a subset of 50 transcripts. These transcripts contain conversations between convicted predators and volunteers posing as children. The posts were extracted from all transcripts for all screen names (the predators and victims sometimes changed screen names). The timeframes represented by the transcripts varied from a few hours to several months. The transcripts ranged in length from 83 lines to 12,704 lines.

3.1 Truth Set Development

All posts by a predator for all 50 transcripts were manually classified by two trained analysts (students in Media and Communication Studies). Each analyst was assigned to a subset of the transcripts and three transcripts were manually coded by both analysts. The overlapping transcripts were used to ensure that the analysts were making near-identical classification decisions. The numbers 200, 600, and 900 were assigned based on the analyst's opinion of the communication being represented. Unclassified lines were assumed to have a classification of 000. Due to file formatting issues, only 33 of these transcripts could be used to generate truth sets for our machine learning algorithms. We used these truth sets to identify the accuracy of our rule-based approach, as well as for development and testing of our machine learning model.

3.2 Rule-based approach to classification

In Section 4 we describe our experiments and compare our decision tree and instance-based learning models to both the truth set and to the rule-based classifier. The rule-based classifier was designed to improve upon a phrase-matching system and our early results showed that the rule-based system (ChatCoder 2) was significantly better than the phrase-matching version (ChatCoder 1) [4]. The ChatCoder 2 rules are based on an expanded dictionary of terms that sorts the words of a post into categories. The rules described below give examples of the terms that appear in our dictionary for each category.

ChatCoder 2 attempts to classify a post as personal information (200) when basic information about the victim or predator's physical location, non-screen names, phone numbers and addresses, or personal photos are exchanged or discussed. Additionally, conversation regarding general likes and dislikes in a non-sexual context, such as "what do you like to do?", was labeled as personal information. Arrangements to meet were explicitly excluded from this category because they would be labeled as approach (900).

Discussing or soliciting non-sexual feelings and attitudes, be they positive, negative, or neutral, about the victim or predator's platonic, familial, and/or romantic relationships was also labeled as personal information. If language related to building, maintaining, dismantling, or ambivalence toward their relationship or the victim's relationship with others was present, the line was labeled as personal information.

Finally, discussion involving the exchange of information about activities, hobbies, favorite musicians, favorite movies, etc. belongs in the personal information class.

Lines are labeled personal information (200) if they follow these patterns:

- A post contains an approach noun (car, hotel), a relationship noun (boyfriend, date), and does not contain a personal information noun (age, pic).
- A post contains a personal information noun and either an action verb (think, do) or a question word (when, who).
- A post contains a relationship noun and either an approach verb (come, see) or an action verb.
- A post contains an activities noun (music, movie)

Grooming (class = 600) is defined as the use of vulgar sexual terms or the discussion of sexual acts and experimentation. ChatCoder 2 classifies a post as grooming if the post contains blatant vulgar language, any discussion or demonstration of sexual acts, innuendos, references to arousal or sex such as "doing things," and the exchange of sexual photos.

ChatCoder 2 also labels a post as grooming when the discussion involves reframing, or the redefinition of sexual behaviors into non-sexual terms, such as connecting sexual acts to messing around, practicing, or teaching.

Grooming represents the bulk of labeled lines and the largest dictionaries of categories. Because there are certain words that indicate grooming regardless of context, there is a category called communicative desensitization word that indicates that a post needs to be labeled as grooming.

Lines are labeled grooming (600) if they follow these patterns:

- A post contains a communicative desensitization word (penis, sex).
- A post contains a communicative desensitization noun (bra, orgasm) and either an action verb or communicative desensitization verb (kiss, suck).
- A post contains a communicative desensitization verb as well as either a second person pronoun or a question word.
- A post contains a communicative desensitization adjective (horny, naked) and either a first or second person pronoun or an action verb.

- A post contains a reframing verb (teach, practice) and either a first or second person pronoun.

Approach (class = 900) is defined as any attempt by the predator to meet the victim in person. ChatCoder 2 tried to classify a line as approach if there was an attempt to speak to the victim over the phone, acquire the victim's physical address or phone number, or bring the victim items. Thus approach indicates a change in aggression on the part of the perpetrator.

A post was also labeled as approach when the predator made an attempt to isolate the victim from his or her support network of family, friends, etc. Any discussion of the physical location of the victim's family and friends was labeled approach, as was any attempt to encourage the victim to lie or conceal things.

Lines are labeled approach (900) if they follow these patterns:

- A post contains an approach verb (come, meet), does not have an information noun, and has a first person pronoun, a second person pronoun, or an approach noun (car, hotel).
- A post contains a family noun (mom, divorce), and an approach verb, an action verb, or an isolation adjective (alone, lonely).
- A post contains an isolation adjective and a second person pronoun.

3.3 Machine Learning Attributes

The dictionary used for ChatCoder 2 was also used to develop the attributes for input to the machine learning algorithm. For all predator posts in all 33 transcripts, we extracted the following information:

- Total number of words in a line (words were defined to be strings of characters separated by white space)
- Number of first person pronouns in a line (e.g. I, me)
- Number of second person pronouns in a line (e.g. you, your)
- Number of third person pronouns in a line (e.g. he, them)
- Number of personal information nouns (e.g. age, pic)
- Number of relationship nouns (e.g. boyfriend, date)
- Number of activities nouns (e.g. movie, favorite)
- Number of family nouns (e.g. mom, sibling)
- Number of communicative desensitization verbs (e.g. kiss, suck)
- Number of communicative desensitization nouns (e.g. bra, orgasm)
- Number of communicative desensitization adjectives (e.g. horny, naked)
- Number of communicative desensitization words (e.g. sex, penis)
- Number of reframing verbs (e.g. teach, practice)
- Number of approach verbs (e.g. meet, see)
- Number of approach nouns (e.g. hotel, car)

Machine learning algorithms are likely to split numeric values differently from nominal values, and we decided to test if the presence of a certain category was as effective as the quantity of terms in that category in a given line; therefore, we converted the "number of" attributes to booleans (Yes, if number is > 0; No, otherwise) and attempted to use supervised learning methods to develop a model using the booleans as well as using the numeric data.

Finally, the class (200,600,900,000) was determined by the truth set (hand classified by trained analysts) and was included as the final attribute in the data file that was used as input to the machine learning tool.

4. Experiments and results

In this section we describe the tools we used and the experimental setup. We ran several sets of experiments on each of the 33 transcripts. Before we can describe the experiments and results, we need to identify appropriate metrics for comparison between systems.

4.1 Metrics

Intercoder reliability compares the number of agreements between two coders doing content analysis. Holsti's method [2] is a commonly used, basic measure. The formula is shown in Equation 1, where *m* is the number of matches, *n1* is the number of lines coded by coder 1, and *n2* is the number of lines coded by coder 2.

$$\text{Reliability} = 2m / (n1 + n2) \qquad (1)$$

Intercoder coder reliability for two trained undergraduate student coders on a set of ten short transcripts ranged from 56.47% to 79.17%.

Intercoder reliability was not a valid metric for the experiments we are using in this paper. Here we are assuming that the human coding is correct and we are trying to match the human coding using either a rule-based or a supervised-learning system. Since we are not interested in ranking our attributes, Area Under the ROC (Receiver Operating Characteristic) Curve (AUC) [14] was not a suitable metric, either. Thus, we have chosen accuracy as a more appropriate metric. Accuracy is computed using Equation 2, where *m* is the number of times the computer system matches the human coding and *N* is the number of predator posts in the transcript.

$$\text{Accuracy} = m/N \qquad (2)$$

Accuracy ranges from 0% to 100% with 100% indicating the computer matches the human coding perfectly.

4.2 Comparison of rule-based to human coding

One of our primary goals was to determine if the hand-coded rules were sufficient to capture the coding rules, or if the machine learning algorithms could do a better job of identifying the class for each line. Thus, our first experiment was a comparison of the rule-based system (ChatCoder 2) to the hand-coded system. We classified the predator lines for each of the 33 transcripts using our rule-based system and computed the accuracy (see Table 1).

The overall accuracy of the rule-based system ranged from 51.95% to 83.90%, and the average is shown to be 68.11%. We ran a comparison between the two human coders (both undergraduate students), and a coding expert (a Professor of Communication Studies) on a test set of 10 short transcripts. One of the coders achieved an overall accuracy of 80.40%; the accuracy for the other coder was 83.43%.

ChatCoder 2 performs best in the grooming category with an average accuracy of 42.22% and a median of 41.45% (see Table 1). Individual chat logs range from 19.48% to 75.00% accuracy when compared to a human coder. This is in part because the majority of the rules are designed to detect grooming, but also in part because the language that a predator uses to desensitize a child is more readily predictable than the language used for approaching or gathering personal information.

Personal information presents the largest problem for ChatCoder 2, because discussion involving personal information presents a very diverse body of conversation and vocabulary. Personal information scores an average accuracy of 15.60%, a median accuracy of 14.29%, and a range from 0 – 44.44% on individual chat logs. Asking the victim to discuss his or her favorite movie is easy to detect, as is asking for pictures or relationship information, but when discussion turns to very specific topics, it becomes difficult for ChatCoder 2 to properly label posts as personal information. For example, without teaching ChatCoder 2 every city and state in the world, ChatCoder will rarely be able to correctly label lines that pertain to a specific location, such as “im in Orange County.”

ChatCoder 2 also encounters problems identifying posts in the approach category, in part because the rules that ChatCoder 2 uses to find approach are, at times, vague, and in part because approach strategies vary greatly from predator to predator. Unlike sexual discussion, which is nearly always guaranteed to contain certain terms and exclude others, discussion involving calling or meeting the victim in person utilizes a much wider range of words and phrases. For instance, to a human coder, the phrase “I can find you without any problem,” in the context of the conversation, should clearly be labeled as approach, but this context is difficult to present to ChatCoder 2 in a rule-based format. ChatCoder 2 was able to accurately identify approach correctly 33.66% of the time on average (median was 33.97%). The accuracy for individual chat logs ranges from 0% to 60.00%.

Table 1: Accuracy for Human vs. Rule-based system for 34 test transcripts

	Accuracy				
Transcript	**Overall**	**000**	**200**	**600**	**900**
Asian_kreationz	83.90	94.80	7.14	31.43	21.95
Aticloose	51.95	84.38	0.00	30.77	38.46
corazon23456partio23456	75.58	84.66	43.75	41.67	53.85
crazytrini85	54.37	80.00	0.00	39.02	40.00
cuteguyinoc2002	74.41	84.76	31.82	56.25	22.22
dble_d1	62.23	79.79	27.78	20.00	39.02
fightingfalconsguy	75.13	86.38	16.67	36.67	35.90
fredold_2000	81.00	91.10	28.57	35.83	33.97
hiexcitement	64.00	81.33	8.33	26.67	43.26
holdyoucloser2003	72.73	87.33	9.09	43.48	50.00
icepirate53	70.74	82.96	20.00	69.23	33.33
italianlover37	54.55	93.33	0.00	50.00	25.00
jkspeedster0112	75.90	86.48	44.44	51.85	0.00
jon_raven2000	71.30	95.89	0.00	31.25	53.85
lee_greer74	70.24	90.38	22.86	19.48	26.15
m4pixeleen	64.68	79.46	21.43	39.02	23.64
marc_00_48089	63.71	82.89	25.00	39.22	36.36
mikespikegetiingcrazytocu	60.91	84.52	7.14	42.86	13.21
netbuckeye	74.74	82.30	16.67	30.00	43.33
nickpaul19802000	77.53	84.62	26.32	54.17	41.67
pavlov1234	74.68	88.48	0.00	31.82	25.00
pitbulldavid2001	67.16	70.63	17.65	39.62	43.33
rayray121980	73.91	86.44	8.33	44.83	25.00
sebastian_calif	59.11	83.80	12.50	41.45	32.43
shinelfmc2005	60.12	77.07	13.33	50.00	46.15
spongebob_giantdick	68.25	82.50	11.11	75.00	60.00
stylelisticgrooves	73.58	85.94	16.67	50.00	35.00
sugardavis	75.36	92.57	14.29	44.05	19.05
sweet_jason002	57.47	73.06	23.53	34.88	15.62
texassailor04	61.18	75.79	29.41	44.00	33.33
the_third_storm	62.03	75.31	0.00	38.78	30.61
user194547	68.04	79.66	0.00	66.67	53.85
vipper_131	67.12	92.52	11.11	43.24	16.36
Average	**68.11**	**84.28**	**15.60**	**42.22**	**33.66**

Interestingly, ChatCoder 2 identifies lines with zero coded categories the best, with an average accuracy of 84.28%. The median is 84.52% and the range is 70.63 to 95.89%. Although ChatCoder 2 has difficulty identifying the correct category for a post, it does a good job of accurately detecting posts that should not be labeled. This statistic is interesting in that it shows that while ChatCoder 2 may not always succeed at making a positive identification, it weeds out those posts that do not contribute to the body of predatory conversation. Knowing which lines belong in the realm of predation is useful, as it helps us better focus our attention on those lines which need better rules or more specific vocabulary.

Currently ChatCoder 2 has no capacity to detect ages, phone numbers, or addresses, which is a problem when attempting to correctly label a post as personal information or approach. It also makes extracting biographical information about the predator difficult, as it quickly becomes taxing to pour through hundreds upon thousands of lines of chat logs for phone numbers. ChatCoder 2 is also particularly weak when context is necessary to correctly code a line, since often the predator will follow a post that is clearly labeled into one class with a contextually based comment that is meaningless to ChatCoder 2. The line "I will show u," referring to sexual acts, and the line "everything," which also refers to sexual acts, should both be labeled as grooming, but the second line gets mislabeled because it is difficult to teach a computer the contextual linguistic capabilities of a human being.

4.3 Machine learning methodology

Once our attributes were identified and extracted from the chat logs, we ran several sets of experiments using the Weka data mining tool kit [13]. We were interested in both decision trees and instance-based learning for classification, thus the C4.5 decision tree learner (implemented as J48 in Weka), the RIPPER rule-learning algorithm using N minimum occurrences of a given rule (implemented as JRip in Weka), and the IBk (instance-based using k neighbors) were used for classification. Though the Naïve Bayes algorithm initially seemed applicable, it returned disappointingly low accuracy and was thus discarded. We used 10-fold cross validation to test the generated model.

In 10-fold cross validation, the data is divided into 10 equal-sized sections. The train-test process is performed 10 times, each time holding a different section out as the test set. Ultimately, the 10 performance measures are averaged together to give the final performance estimate.

For all of the datasets, the lines that were classified as 000 significantly outnumbered the lines that were classified as 200, 600 and 900. To overcome the machine learning bias toward the large 000 class, we weighted our input data sets. We determined empirically that tripling the 600, repeating the 200 and 900 instances seven times, and using only 20% of the 000 instances obtained the best results.

4.4 Results of learning using individual transcripts

Each of the 33 test files was run using the J48, the JRip with N = 1, the JRip with N = 2, the IBk with k=1, and the IBk with k=3 classifiers in Weka. The percentage of correctly classified instances (using 10-fold cross validation) is reported in Table 2 for the test files that used counts and in Table 3 for the test files that used booleans. The rule-based results are repeated for comparison purposes. The values are highlighted for comparison with the rule-based system.

The values in Tables 2 and 3 clearly show that machine learning can improve upon the rule-based system when working with a specific transcript. Furthermore, the boolean values are slightly better than the counts, and IBk with three nearest-neighbors is not as accurate as IBk with 1 neighbor or the J48 decision tree. Of the algorithms used, JRip is the least effective.

A paired, two-tail Student t-test was performed to determine if the difference between the accuracy of ChatCoder's rule-based approach and the machine learning algorithms was significant when comparing individual chatlogs. While the mean difference between the rule-based approach and the machine learning algorithms may appear insignificant, when the results of the t-test are analyzed, it becomes clear that the difference is significant. With 99% confidence ($\alpha/2 = .005$), we reject the null hypothesis that the rule-based approach is statistically equivalent to the machine learning algorithms when the calculated t-statistic is greater than the corresponding critical value, 2.576. In all cases – IBK, with k=1 and k=3, J48, and JRip with N=1 and N=2 – the t-statistic is considerably larger than the critical value, leading us to the conclusion that the machine learning algorithms provide a significant statistical improvement over the original rule-based method when comparing individual transcripts, as can be seen in Table 4 for count and Table 5 for booleans.

Table 4: mean differences and t-statistics for 34 individual transcripts using counts.

	Rule-based vs. IBK k = 1	**Rule-based vs. IBK k=3**	**Rule-based vs. J48**	**Rule-based vs. JRip N=1**	**Rule-based vs. JRip N=2**
Mean Difference	-15.00	-9.15	-13.36	-7.65	-7.97
t-Statistic	8.63	5.25	7.68	3.84	3.72

Table 5: mean differences and t-statistics for 34 individual transcripts using Booleans.

	Rule-based vs. IBK k = 1	**Rule-based vs. IBK k=3**	**Rule-based vs. J48**	**Rule-based vs. JRip N=1**	**Rule-based vs. JRip N=2**
Mean Difference	-14.83	-8.36	-12.61	-6.66	-6.83
t-Statistic	7.83	4.53	6.58	3.14	3.22

Table 2: Accuracy for individual transcripts using counts

Transcript	Rule-based	IBK k=1	IBK k=3	J48	JRip N=1	JRip N=2
Asian_kreationz	83.90	85.03	78.74	81.59	71.12	72.46
Aticloose	51.95	86.50	73.84	84.39	74.68	73.84
corazon23456partio23456	75.58	86.81	82.01	85.37	81.53	82.01
crazytrini85	54.37	83.21	72.99	78.47	72.68	71.9
cuteguyinoc2002	74.41	76.14	71.02	71.02	69.60	68.75
dble_d1	62.23	80.25	72.63	79.00	64.75	63.38
fightingfalconsguy	75.13	85.59	78.13	81.34	75.99	74.09
fredold_2000	81.00	85.48	82.39	83.36	79.63	79.57
hiexcitement	64.00	81.01	77.49	80.48	74.63	75.7
holdyoucloser2003	72.73	90.19	81.88	89.55	88.06	87.42
icepirate53	70.74	87.66	81.36	85.30	84.25	84.78
italianlover37	54.55	85.37	73.98	82.93	75.61	76.42
jkspeedster0112	75.90	88.59	83.44	85.40	84.91	84.9
jon_raven2000	71.30	86.79	82.50	84.64	85.00	82.86
lee_greer74	70.24	74.72	69.31	73.67	60.25	60.32
m4pixeleen	64.68	85.41	77.26	81.00	75.79	75.91
marc_00_48089	63.71	85.32	78.87	83.39	74.84	75.32
mikespikegetiingcrazytocu	60.91	83.19	78.95	81.66	77.76	77.59
netbuckeye	74.74	77.46	73.64	77.46	60.16	60.76
nickpaul19802000	77.53	90.86	83.43	88.76	86.86	86.48
pavlov1234	74.68	80.39	74.51	75.16	70.59	70.59
pitbulldavid2001	67.16	84.28	76.68	82.56	72.76	72.56
rayray121980	73.91	84.10	74.62	81.35	68.74	73.09
sebastian_calif	59.11	77.56	71.68	75.93	78.15	67.97
shinelfmc2005	60.12	87.66	85.47	86.65	98.19	77.97
spongebob_giantdick	68.25	98.19	92.77	96.99	76.79	97.59
stylelisticgrooves	73.58	84.11	72.86	81.79	67.91	78.21
sugardavis	75.36	78.51	73.93	78.22	76.79	68.34
sweet_jason002	57.47	73.97	67.50	73.72	67.87	67.87
texassailor04	61.18	87.21	78.20	86.63	83.43	84.3
the_third_storm	62.03	87.05	80.69	83.04	78.35	78.91
user194547	68.04	54.68	52.88	51.80	53.96	53.24
vipper_131	67.12	73.66	67.80	71.09	61.32	62.24
Average	**68.11**	**82.94**	**76.47**	**80.72**	**74.94**	**74.77**

Table 3: Accuracy for 34 individual test transcripts using booleans.

Transcript	**Rule-based**	**IBK k=1**	**IBK k=3**	**J48**	**JRip N=2**	**JRip N=1**
Asian_kreationz	83.90	81.50	78.29	81.59	73.05	72.46
Aticloose	51.95	86.5	75.95	82.7	73.84	76.79
corazon23456partio23456	75.58	87.05	82.01	86.33	82.97	82.97
crazytrini85	54.37	82.12	75.55	77.37	72.99	77.26
cuteguyinoc2002	74.41	72.73	67.05	70.17	66.76	66.76
dble_d1	62.23	72.25	79.63	77.13	64.63	65.63
fightingfalconsguy	75.13	83.59	79.45	81.69	74.96	75.48
fredold_2000	81.00	84.54	80.75	83.16	78.6	78.69
hiexcitement	64.00	81.01	77.42	80.15	76.03	75.9
holdyoucloser2003	72.73	89.55	85.07	89.98	86.78	85.93
icepirate53	70.74	87.93	81.36	85.83	85.56	85.38
italianlover37	54.55	85.37	73.98	82.93	76.42	78.05
jkspeedster0112	75.90	88.22	83.07	85.52	85.28	84.79
jon_raven2000	71.30	86.79	82.5	84.64	83.21	83.21
lee_greer74	70.24	72.96	67.42	71.14	58.29	57.51
m4pixeleen	64.68	84.73	79.07	81.79	76.81	77.04
marc_00_48089	63.71	85.32	77.9	83.06	74.35	76.77
mikespikegetiingcrazytocu	60.91	83.19	77.42	82.17	77.93	77.93
netbuckeye	74.74	77.46	72.84	76.26	61.97	59.55
nickpaul19802000	77.53	90.86	84.76	88	86.1	87.24
pavlov1234	74.68	80.39	71.9	74.84	70.92	70.59
pitbulldavid2001	67.16	82.12	75.25	77.54	69.09	68.96
rayray121980	73.91	84.4	74.01	82.57	77.68	78.59
sebastian_calif	59.11	77.89	73.53	77.34	71.57	69.5
shinelfmc2005	60.12	87.57	85.37	86.75	78.79	79.25
spongebob_giantdick	68.25	98.19	90.96	97.59	96.99	96.99
stylelisticgrooves	73.58	84.64	73.75	81.61	78.75	80.18
sugardavis	75.36	78.08	70.63	77.08	65.33	65.76
sweet_jason002	57.47	73.85	67	73.22	65.38	67.12
texassailor04	61.18	87.21	80.81	86.63	84.3	83.14
the_third_storm	62.03	86.38	81.14	84.49	79.8	79.46
user194547	68.04	85.97	78.78	86.69	84.17	84.89
vipper_131	67.12	72.33	64.92	70.47	60.91	60.91
Average	**68.11**	**83.11**	**77.26**	**81.47**	**75.76**	**76.08**

However, the results shown in Tables 2 and 3 are misleading. We are using instances for a single transcript to predict other classifications in the same transcript. This is not what we want to learn. We want to use sample instances to predict unseen instances across transcripts. In the next section we describe our results when we merge the 33 files together into a single input file to be used for training and testing the classification model.

4.5 Results of learning using all classified instances

Figure 1 shows the comparison of the Weka classifiers vs. ChatCoder 2 for the boolean input files compiled into a single input file (Figure 2 shows the corresponding figures for the counts). There were 23,731 instances in the data set, and 10-fold cross evaluation was used to produce the accuracy figures. These figures clearly show that the machine learning algorithms do not significantly improve over the rule-based approach when the data is analyzed as one single transcript.

Using booleans, the C4.5 algorithm (using a confidence factor of .001) generates a pruned tree with 130 leaves. Overall, the tree branches and leaves are either representative of the rules used in ChatCoder 2 or intuitive extensions of the rules. For instance, if Weka finds a communicative desensitization word, an approach noun, and no reframing verb or approach verb, the line is labeled as grooming (600). This represents an extension of pre-existing rules in ChatCoder 2 that we could potentially use to better refine our model.

The number of words in a line attribute was used more often than we expected in the Boolean data. This attribute is frequently used to break ties or assist in correctly labeling an ambiguous line. When deciding between grooming and approach – a decision that often stymies ChatCoder 2 – the C4.5 algorithm uses the number of words in a line. In instances where there is only an approach verb and an activities noun in a post, if the line contains more than seven words, it is labeled as approach (900), and if there are seven or fewer words, then the post is labeled personal information (200). In general, when a limit for the number of words in a line is used, if the line has fewer words, it is labeled as personal information. Thus personal information discussion often involves fewer words than grooming or approach. With this information, we are better informed about the patterns inherent in the linguistic patterns that predators use to speak to minors and, eventually, approach them for sexual interaction.

A tree with 161 leaves is generated by the C4.5 algorithm when counts are used instead of boolean values. The branches of the counts-based tree generally agree with the rules used in ChatCoder 2, but instead of promoting clarity as the boolean tree does, the count-based approach adds complexity, as can be seen in Figure 3. In instances where a simple rule using a single attribute would suffice, the decision tree uses multiple splits to differentiate between classes. For example, though the attribute number of words in a line proved surprisingly useful in the Boolean transcripts, in the count-based approach, it creates unintuitive, confusing results. In one tree, four separate rules based solely on the number of words in a line label lines into the approach category when a single rule would suffice. Despite the fact that the counts provide the highest accuracy figures for approach, we believe our rule-based system is better because the additional complexity in the decision tree indicates that the tree is likely over-fitted to the training set.

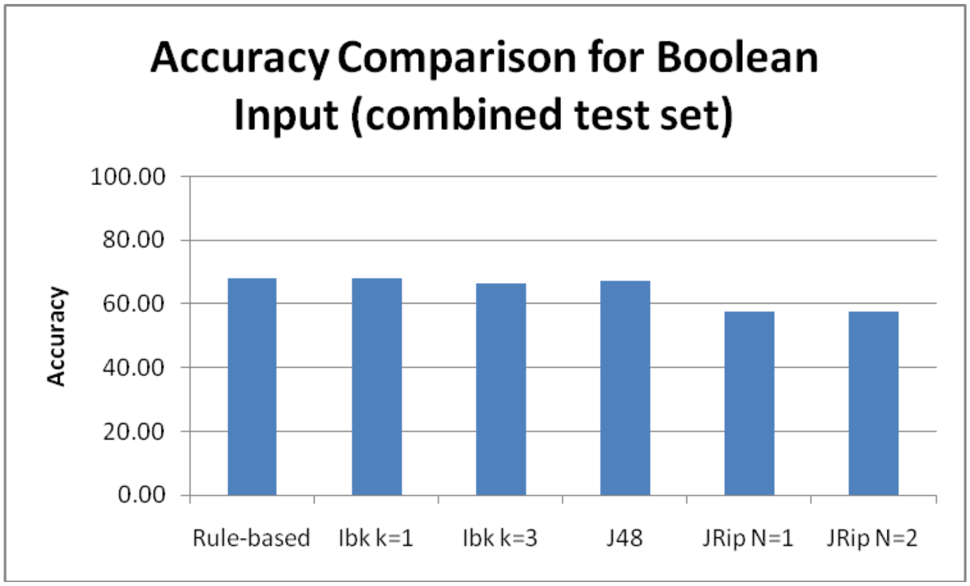

Figure 1: Accuracy comparison for boolean input

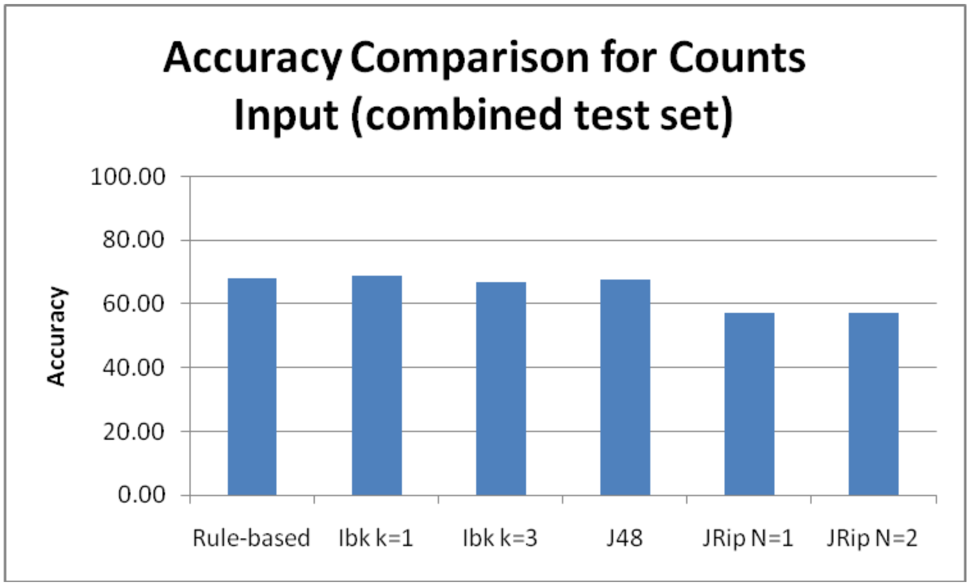

Figure 2: Accuracy comparison for counts input

The if-then-else rules generated by the RIPPER algorithm were many – ranging from 38 to 45 rules. In both cases, counts and booleans, the rules generated by the RIPPER algorithm were overly complicated and unintuitive, much as we experienced with the trees from the J48 algorithm. ChatCoder 2 uses 12 rules compared to the 38 to 45 rules employed by the RIPPER algorithm. For example, ChatCoder 2 labels grooming using five rules, but the RIPPER algorithm uses an average of 29 rules.

```
|   |   2pro > 0
|   |   |   familyn <= 0
|   |   |   |   approachn <= 0
|   |   |   |   |   commadj <= 0
|   |   |   |   |   |   3pro <= 0
|   |   |   |   |   |   |   commvb <= 0
|   |   |   |   |   |   |   |   commn <= 0
|   |   |   |   |   |   |   |   |   infon <= 0
|   |   |   |   |   |   |   |   |   |   relationshipn <= 0
|   |   |   |   |   |   |   |   |   |   |   numWords <= 9
|   |   |   |   |   |   |   |   |   |   |   |   reframingvb <= 0: x200
|   |   |   |   |   |   |   |   |   |   |   |   reframingvb > 0
|   |   |   |   |   |   |   |   |   |   |   |   |   activn <= 0: x600
|   |   |   |   |   |   |   |   |   |   |   |   |   activn > 0: x200
|   |   |   |   |   |   |   |   |   |   |   numWords > 9
|   |   |   |   |   |   |   |   |   |   |   |   reframingvb <= 0: x900
|   |   |   |   |   |   |   |   |   |   |   |   reframingvb > 0
|   |   |   |   |   |   |   |   |   |   |   |   |   numWords <= 11
|   |   |   |   |   |   |   |   |   |   |   |   |   |   1pro <= 0: x600
|   |   |   |   |   |   |   |   |   |   |   |   |   |   1pro > 0: x200
|   |   |   |   |   |   |   |   |   |   |   |   |   numWords > 11: x600
```

Figure 3: C4.5 tree generated by Weka using counts dataset

Below, a sampling of the grooming rules generated by JRip in Weka demonstrates their unintuitive, over-complicated nature.

- communicative desensitization word
- communicative desensitization verb, first person pronoun, number of words ≤ 9, and no approach verb
- a second and third person pronoun, no approach noun, no first person pronoun, and a communicative desensitization noun
- number of words ≥ 7, a communicative desensitization adjective, and no approach verb
- a second and third person pronoun, no approach verb, no approach noun, and number of words ≤ 16
- communicative desensitization verb, a relationship noun, no approach verb, and number of words ≥ 7
- communicative desensitization noun, no approach verb, number of words ≥ 7, and number of words ≤ 8

- communicative desensitization verb, a third person pronoun, no approach verb, no approach noun, number of words ≥ 10, and number of words ≤ 11
- communicative desensitization noun, a third person pronoun, no approach verb, number of words ≥ 9, and number of words ≥ 29

ChatCoder 2 uses a more cohesive, efficient set of rules. Figures 1 and 2 show that the additional complexity employed by the tree and rule learning algorithms does not dramatically improve the results we obtained using 12 rules.

5. Conclusions

In this paper we have validated the accuracy of our rule-based system for labeling predatory posts in a chat transcript. We have shown that standard machine learning techniques are not sufficient to improve upon our current system. In fact, the C4.5 decision tree model adds complexity without improving reliability.

We have also discovered an additional attribute, number of words in a line, which may be used to refine our system. We do a reasonable job of determining which lines should be labeled as predatory; the accuracy for class 000 in the boolean group file is 75.13%.

It is important to note that, although we would prefer to improve our numbers (particularly with regards to the approach category) there is always disagreement between human coders. It is likely our accuracy figures are lower because we are finding and labeling lines that the human coders missed. This is particularly true for long transcripts, when humans are likely to become fatigued after reading pages of chat dialog. On the other hand, a computerized system, like ChatCoder, will provide consistency in labeling.

One obvious avenue for future work lies in identifying and capturing context. We currently use only the information contained in a single post to identify its label, but we have seen many examples where the lines that appear before a given line greatly influence the human coding decision (and justifiably so). Incorporating a window of text processing, as well as information pertaining to victim response (as we currently only look at the predator posts) is an important next step.

Acknowledgements:

This material is based in part upon work supported by the National Science Foundation under Grant No. 0916152. Additional funding was provided by the Ursinus Summer Fellows program. Our thanks extend to the many students and colleagues at Ursinus College who have provided input to this project, as well as to the anonymous reviewers for their helpful comments.

References:

1. Adams, P. H. and Martell, C.H. Topic detection and extraction in chat. In *Proceedings of the 2008 IEEE International Conference on Semantic Computing.* 2008. 581-588.
2. Holsti, O. R. *Content Analysis for the Social Sciences and Humanities.* Reading, MA: Addison-Wesley, 1969.
3. Hughes, D.; Rayson, P.; Walkerdine, J.; Lee, K.; Greenwood, P.; Rashid, A.; May-Chahal, C.; and Brennan, M. Supporting law enforcement in digital communities through natural language analysis. In *Proceedings of the Second International Workshop on Computational Forensics (IWCF'08).* 2008.
4. Kontostathis, A.; Edwards, L.; and Leatherman, A. ChatCoder: Toward the Tracking and Categorization of Internet Predators. In *Proceeding of the Text Mining Workshop 2009 held in conjunction with the Ninth SIAM International Conference on Data Mining (SDM 2009).* Sparks, NV, May 2009.
5. Kontostathis, A.; Edwards, L.; and Leatherman, A. Text Mining and Cybercrime. In *Text Mining: Applications and Theory*. Berry, M.W. and Kogan, J. (eds). John Wiley & Sons, Ltd, 2009.
6. Kontostathis, A.; Edwards, L.; Bayzick, J.; McGhee, I.; Leatherman, A.; and Moore, K. Comparison of Rule-based to Human Analysis of Chat Logs. In *Proceedings of the first International Workshop on Mining Social Media (MSM09).* Seville, Spain, November 2009.
7. Leatherman, A. Luring language and virtual victims: Coding cyber-predators on-line communicative behavior. Media and Communication Studies, Ursinus College, Collegeville, PA, 2009.
8. NCMEC. *National center for missing and exploited children.* (accessed April 2010) http://www.missingkids.com/en US/documents/CyberTiplineFactSheet.pdf (accessed January 2010).
9. Olson, L.; Daggs, J; Ellevold, B.; and Rogers, T. Entrapping the innocent: Toward a theory of child sexual predators' luring communication. *Communication Theory* 17, no. 3 (2007): 231-251.
10. Pendar, N. Toward spotting the pedophile: Telling victim from predator in text chats. *Proceedings of the First IEEE International Conference on Semantic Computing.* 2007.
11. PJ. *Perverted Justice.* www.perverted-justice.com (accessed November 2009).
12. Riffe D.; Lacy, S.; and Fico, F. *Analyzing Media Messages: Using Quantitative Content Analysis in Research.* Lawrence Erlbaum Associates, 1998.
13. Witten, E. and Frank, I. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann Publishers, 2005.
14. Wu, S. and Flach, P.A. Scored and Weighted AUC Metrics for Classifier Evaluation and Selection, *Second Workshop on ROC Analysis in Machine learning, ROCML'05*. 2005.
15. Yin, D.; Xue, Z.; Hong, L.; Davison, B.D.; Kontostathis, A.; and Edwards, L. Detection of Harassment on Web 2.0. In *CAW 2.0 '09: Proceedings of the 1st Content Analysis in Web 2.0 Workshop,* Madrid, Spain, 2009.