# Using Machine Learning to Detect Cyberbullying

Kelly Reynolds, April Kontostathis
Mathematics and Computer Science Department
Ursinus College
Collegeville PA 19426
Email: kereynolds,akontostathis@ursinus.edu

Lynne Edwards
Media and Communication Studies Department
Ursinus College
Collegeville PA 19426
Email: ledwards@ursinus.edu

***Abstract*—Cyberbullying is the use of technology as a medium to bully someone. Although it has been an issue for many years, the recognition of its impact on young people has recently increased. Social networking sites provide a fertile medium for bullies, and teens and young adults who use these sites are vulnerable to attacks. Through machine learning, we can detect language patterns used by bullies and their victims, and develop rules to automatically detect cyberbullying content.**

**The data we used for our project was collected from the website Formspring.me, a question-and-answer formatted website that contains a high percentage of bullying content. The data was labeled using a web service, Amazon's Mechanical Turk. We used the labeled data, in conjunction with machine learning techniques provided by the Weka tool kit, to train a computer to recognize bullying content. Both a C4.5 decision tree learner and an instance-based learner were able to identify the true positives with 78.5% accuracy.**

## I. Introduction

Social networking sites are great tools for connecting with people. However, as social networking has become widespread, people are finding illegal and unethical ways to use these communities. We see that people, especially teens and young adults, are finding new ways to bully one another over the Internet. Close to 25% of parents in a study conducted by Symantec reported that, to their knowledge, their child has been involved in a cyberbullying incident [1].

There are no well-known datasets for research on cyberbullying. A set of large datasets was made available at the Content Analysis on the Web 2.0 (CAW 2.0) workshop for a misbehavior detection task, however this dataset was unlabeled (i.e. it is not known which posts actually contain cyberbullying). Furthermore, the data was pulled from a variety of sources, and, with the exception of the data from Kongregate (a gaming website), the datasets appear to be discussions among adults. In order to conduct the study reported herein, we developed our own labeled dataset containing data from a webcrawl of Formspring.me. Formspring.me was chosen because the site is populated mostly by teens and college students, and there is a high percentage of bullying content in the data. The collection and labeling of this data is described in detail in Section III.

A variety of lexical features were extracted from the Formspring.me post data, and several data mining algorithms that are available in the Weka toolkit were used to develop a model for the detection of cyberbullying. When comparing our results, we were focused more on recall than on precision. In other words, we were willing to accept some false positives in order to increase the percentage of true positives that could be identified by the tool. Our machine learning experiments are described in Section IV and the results appear in Section V.

## II. Background and Related Work

Patchin and Hinduja define cyberbullying as willful and repeated harm inflicted through the medium of electronic text [2]. A review of the adolescent psychology literature reveals nine different types of cyberbullying that can be distinctly identified [3], [4], [5].

Very few other research teams are working on the detection of cyberbullying. As mentioned earlier, a misbehavior detection task was offered by the organizers of CAW 2.0, but only one submission was received. Yin, et. al determined that the baseline text mining system (using a bag-of-words approach) was significantly improved by including sentiment and contextual features. Even with the combined model, a support vector machine learner could only produce a recall level of 61.9% [6].

A recent paper describes similar work is that is being conducted at Massachusetts Institute of Technology. The research is aimed towards detecting cyberbullying through textual context in YouTube video comments. The first level of classification is to determine if the comment is in a range of sensitive topics such as sexuality, race/culture, intelligence, and physical attributes. The second level is determining what topic. The overall success off this experiment was 66.7% accuracy for detecting instances of cyberbullying in YouTube comments. This project also used a support vector machine learner [7].

## III. Data Collection

In this section we describe the collection and labeling of the data we used in our experiments.

### A. Dataset Origin

The website Formspring.me is a question and answer based website where users openly invite others to ask and answer questions. What makes this site especially prone to cyberbullying is the option for anonymity. Formspring.me allows users to post questions anonymously to any other user's page.

To obtain this data, we crawled a subset of the Formspring.me site and extracted information from the sites of

18,554 users. The users we selected were chosen randomly. The number of questions per user ranged in size from 1 post to over 1000 posts. We also collected the profile information for each user.

We were interested in first developing a language-based model for identifying cyberbullying, and the only fields we used in our study were the text of the question and the answer. Clearly a lot of rich information has been collected, and we hope to expand our experiments in future work to take advantage of the profile information.

### B. Labeling the data

We extracted the question and answer text from the Formspring.me data for 10 files for the training set and 10 files for the testing set. These files were chosen randomly from the set of 18,554 users that were crawled, but we ensured that there was no overlap between the two sets of files. We used the same procedure to identify class labels both the training and the testing sets.

We decided to use Amazon's Mechanical Turk service to determine the labels for our truth sets. Mechanical Turk is an online marketplace that allows requestors to post tasks (called HITs) which are then completed by paid workers. The workers are paid by the requestors per HIT completed. The process is anonymous (the requestor cannot identify the workers who answered a particular task unless the worker chooses to reveal him/herself). The amount offered per HIT is typically small. We paid three workers .05 cents each to label each post. Our training set contained 2696 posts; our test set contained 1219 posts.

Each HIT we posted displayed a Question and Answer from the Formspring crawl and a web form that requested the following information:

1) Does this post contain cyberbullying (Yes or No)?
2) On a scale of 1 (mild) to 10 (severe) how bad is the cyberbullying in this post (enter 0 for no cyberbullying)?
3) What words or phrases in the post(s) are indicative of the cyberbullying (enter n/a for no cyberbullying)?
4) Please enter any additional information you would like to share about this post.

We asked three workers to label each post because the identification of cyberbullying is a subjective task. Our class labels were "yes" for a post containing cyberbullying and "no" for a post without cyberbullying. The data provided by the other questions will be used for future work. At least two of the three workers had to agree in order for a post to receive a final class label of "yes" in our training and testing sets.

Of the 2696 posts in the training set, 196 received a final class label of "yes," indicating the presence of cyberbullying (7.2% of the posts). Of the 1219 posts in our test set, 173 were identified as cyberbullying (14.2%), almost twice as many. These ratios confirmed our suspicion that the percentage of cyberbullying in the Formspring data was much higher than in other datasets that we've seen.

## IV. Developing the Model

Machine Learning is the process of training a computer to predict a label using a set of attributes and a truth set [8]. The machine learning tool then evaluates the success of the model and produces statistical results indicating the success of the learning experiment.

### A. Developing Features for Input

As discussed earlier, we wanted to develop a model based on textual features. This section describes the identification and extraction of features from each Formspring post. We were determined to avoid a bag-of-words approach for several reasons. First, the feature space with a bag-of-words approach is very large. Second, we wanted to be able to reproduce the model in code, and having each term as a feature would make that impractical. Third, we wanted to be able to understand *why* a post was considered as containing cyberbullying as this will inform the development of a communicative model for cyberbullying detection.

One thing was clear from the labeling project, there are "bad" words that make a post more likely to be labeled as cyberbullying. In order to leverage this information, we identified a list of insult and swear words, posted on the website www.noswearing.com. This list, containing 296 terms, was downloaded and each word on the list was given a severity level by our team. The levels were 100 (ex. butt, idiot), 200 (ex. trash, prick), 300 (ex. asshole, douchebag), 400 (ex. fuckass, pussy), and 500 (ex. buttfucker, cuntass). The classification of these terms into severity levels was subjective and will be reviewed in future work.

We were interested in both the number of "bad" words (NUM) and the density of "bad" words (NORM) as features for input to the learning tool. We therefore extracted two different training sets, one containing the count information, and one containing normalized information. We normalized by simply dividing the number of words at each severity level by the total number of words in the post, and then multiplying by 100 to get an integer value (for example, if there were 6 100-level words in a 10 word post, the 100-level would be reported as 60).

We also generated a feature to measure the overall "badness" of a post. We call this feature SUM and computed it by taking a weighted average of the "bad" words (weighting by the severity assigned).

The SUM and TOTAL features were included in both the NUM and the NORM versions of our datasets. The class label (YES, NO) was also extracted from the Mechanical Turk file and included in the input to the machine learning tool.

### B. Learning the Model

Weka is a software suite for machine learning that creates models using a wide variety of well-known algorithms [8]. We identified the following algorithms as most useful for our project.

- J48: The J48 option uses the C4.5 algorithm to create a decision tree model from the attributes provided [9].

TABLE I
THE TP PERCENTAGES FOR THE NUM TRAINING SET USING J48, JRIP, IBK, AND SMO ALGORITHMS

| NUM Data Set | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | Description | Number of Repetitions | | | | | | | | | |
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| J48 | % correctly labeled positive | 54.4 | 63.6 | 72.3 | 75.6 | 76.1 | 76.5 | 77.7 | 78.2 | 78.5 | 78.5 |
| | # of Leaves | 14 | 11 | 7 | 2 | 2 | 2 | 5 | 6 | 26 | 26 |
| | size of tree | 27 | 21 | 13 | 3 | 3 | 3 | 9 | 11 | 51 | 51 |
| JRIP | % correctly labeled positive | 56.9 | 70.0 | 75.6 | 75.1 | 76.1 | 76.0 | 76.8 | 76.9 | 77.0 | 77.3 |
| | # of Rules | 3 | 6 | 6 | 2 | 2 | 5 | 7 | 4 | 3 | 4 |
| IBK1 | % correctly labeled positive | 55.4 | 75.4 | 78.1 | 78.1 | 78.5 | 78.5 | 78.5 | 78.5 | 78.5 | 78.5 |
| IBK3 | % correctly labeled positive | 54.9 | 62.3 | 74.5 | 77.4 | 78.5 | 78.5 | 78.5 | 78.5 | 78.5 | 78.5 |
| SMO | % correctly labeled positive | 36.9 | 53.8 | 61.3 | 61.5 | 62.2 | 63.1 | 65.0 | 66.7 | 67.2 | 67.2 |

TABLE II
THE TP PERCENTAGES FOR THE NORM TRAINING SET USING J48, JRIP, IBK, AND SMO ALGORITHMS

| NORM Data Set | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | Description | Number of Repetitions | | | | | | | | | |
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| J48 | % correctly labeled positive | 53.3 | 63.3 | 73.0 | 75.8 | 76.5 | 77.6 | 78.3 | 78.5 | 78.5 | 78.5 |
| | # of Leaves | 11 | 12 | 18 | 14 | 11 | 13 | 13 | 13 | 6 | 6 |
| | size of tree | 21 | 23 | 35 | 27 | 21 | 25 | 25 | 25 | 11 | 11 |
| JRIP | % correctly labeled positive | 57.4 | 66.4 | 74.5 | 75.4 | 75.6 | 75.9 | 76.6 | 77.2 | 77.5 | 77.0 |
| | # of Rules | 4 | 4 | 2 | 2 | 6 | 3 | 3 | 3 | 3 | 3 |
| IBK1 | % correctly labeled positive | 54.4 | 75.4 | 78.1 | 78.1 | 78.5 | 78.5 | 78.5 | 78.5 | 78.5 | 78.5 |
| IBK3 | % correctly labeled positive | 52.8 | 65.5 | 76.1 | 77.7 | 78.5 | 78.5 | 78.5 | 78.5 | 78.5 | 78.5 |
| SMO | % correctly labeled positive | 46.2 | 54.1 | 59.0 | 61.2 | 63.4 | 63.9 | 64.5 | 65.1 | 65.2 | 67.5 |

When working with decision trees, it is important to consider the size of the tree that is generated, as well as the accuracy of the model. A large, complex tree may be overfitted to the data. A small, simple tree may indicate that the training set is not well balanced and the model cannot be clearly identified.

- JRIP: JRIP is a rule based algorithm that creates a broad rule set then repeatedly reduces the rule set until it has created the smallest rule set that retains the same success rate [10].
- IBK: The instance-based (IBK) algorithm implemented in Weka is a k-nearest neighbor approach [11]. We used the IBK method with $k = 1$ and $k = 3$.
- SMO: We wanted to use a support vector machine algorithm for testing also. Other teams that are working on similar projects found reasonable success with support vector machines [6], [7] . The SMO algorithm in Weka is a function-based support vector machine algorithm based on sequential minimal optimization [12]. In Section V we show that SMO was the least successful algorithm for our experiments.

### C. Class Weighting

Less than 10% of the training data is positive (contained cyberbullying). As a result, the learning algorithms, by default, generated a lot of false negatives (i.e. they can reach accuracy figures of over 90% by almost ignoring the cyberbullying examples). As discussed earlier, we are interested primarily in recall. We would prefer to have innocent posts labeled as cyberbullying (false positives) instead of mislabeling cyberbullying posts as innocent (false negatives).

In order to overcome the problem of sparsicity in the positive instances, we increased the weight of these instances in the dataset. We did this by simply copying the positive training examples multiple times in order to balance the training set and provide an incentive for the learners to identify the true positives. The results of the weighting experiments are described in Section V.

### D. Evaluation

We used two evaluation approaches in our experiments. As described in Section III we developed and labeled an independent test set using the same procedure that was used in the development of our training set. This set was used for testing of our most successful algorithm and the results appear below. However, the characteristics of the test set appear to be significantly different than from the training set. More than twice as many posts were identified as cyberbullying. Additionally, both sets are relatively small and contain data from only 10 users of Formspring.me. For this reason we also report statistics from experiments using cross validation.

Cross validation is an approach used to evaluate learning algorithms when the amount of labeled data available is small [8]. Ten-fold cross validation is considered to be the standard

TABLE III
ACCURACY FIGURES FOR J48 WITH MULTIPLE WEIGHTINGS

| NORM Data Set | | |
|---|---|---|
| Weighting Applied | True Positive Accuracy | Overall Accuracy |
| 8 | 61.6% | 81.7% |
| 9 | 67.4% | 78.8% |
| 10 | 67.4% | 78.8% |

approach to evaluation in many machine learning experiments, and we use this metric when we report the results for our experiments in the next section.

## V. RESULTS

In this section we describe and discuss our ability to accurately predict instances of cyberbullying in Formspring.me data.

We compare the results using the NUM training set to the NORM training set in Tables I and II, respectively. These tables report the recall for identifying cyberbully using 10-fold cross validation. We see that NORM training set generally outperforms the NUM training set for all repetitions and for all algorithms, with the exception of the SMO algorithm. We also see that as the weighting of positive instances increases, the NORM success rates are slightly higher than the NUM success rates. We conclude from these results that the percentage of "bad" words in a post is more indicative of cyberbullying than a simple count.

Table II shows the improvement in accuracy using the J48 algorithm for the NORM data set (10-fold cross validation) as we increase the percentage of positive instances in the training set. We want a model that maximizes the true positive success rate, but also creates a decision tree that is neither too complex nor too simple. We see in the NORM chart that duplicating the positive instances 7 and 8 times creates a decision tree that has 13 leaves, but the 9 and 10 weightings produce trees with only six leaves. Repeating the instances 8 times produces the best balance between accuracy and tree size.

As mentioned in Section IV we also developed an independent test set to evaluate the model. When testing the model created with the 8 weight NORM training set and J48 algorithm, we obtained a true positive accuracy of 61.6% and an overall accuracy of 81.7% (see Table III). Interestingly, the smaller tree produced by the 9 and 10 repetition data seems to perform better with the independent test set. A close analysis of both trees tells an interesting story. The smaller tree relies only on the SUM and the TOTAL WORDS features. The larger one also relies on the percentage of 100-level words in the post. It seems counter-intuitive that the 100-level words are most indicative of cyberbullying. Perhaps those words are just used more commonly than some of the esoteric insults that appear at the 500 level.

## VI. CONCLUSIONS

In this paper, we used a language-based method of detecting cyberbullying. By recording the percentage of curse and insult words within a post, we were able to correctly identify 78.5% of the posts that contain cyberbullying in a small sample of Formspring data. Our results indicate that our features do a reasonable job of identifying cyberbullying in Formspring posts and also that there is plenty of room for improvement on this timely and important application of machines learning to web data.

## ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant No. 0916152. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## REFERENCES

[1] D. Poeter. (2011) Study: A Quarter of Parents Say Their Child Involved in Cyberbullying. pcmag.com. [Online]. Available: http://www.pcmag.com/article2/0,2817,2388540,00.asp

[2] J. W. Patchin and S. Hinduja, "Bullies move Beyond the Schoolyard; a Preliminary Look at Cyberbullying," *Youth Violence and Juvenile Justice*, vol. 4, no. 2, pp. 148–169, 2006.

[3] Anti Defamation League. (2011) Glossary of Cyberbullying Terms. adl.org. [Online]. Available: http://www.adl.org/education/curriculum_connections/cyberbullying /glossary.pdf

[4] N. E. Willard, *Cyberbullying and Cyberthreats: Responding to the Challenge of Online Social Aggression, Threats, and Distress*. Research Press, 2007.

[5] D. Maher, "Cyberbullying: an Ethnographic Case Study of one Australian Upper Primary School Class," *Youth Studies Australia*, vol. 27, no. 4, pp. 50–57, 2008.

[6] D. Yin, Z. Xue, L. Hong, B. D. Davison, A. Kontostathis, and L. Edwards, "Detection of Harassment on Web 2.0," in *Proc. Content Analysis of Web 2.0 Workshop (CAW 2.0)*, Madrid, Spain, 2009.

[7] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the Detection of Textual Cyberbullying," in *Proc. IEEE International Fifth International AAAI Conference on Weblogs and Social Media (SWM'11)*, Barcelona, Spain, 2011.

[8] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. San Francisco, CA: Morgan Kauffman, 2005.

[9] R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kauffman, 1993.

[10] W. W. Cohen, "Fast Effective Rule Induction," in *Proc. Twelfth International Conference on Machine Learning (ICML'95)*, Tahoe City, CA, 1995, pp. 115–123.

[11] D. W. Aha and D. Kibler, "Instance-based Learning Algorithms," *Machine Learning*, vol. 6, pp. 37–66, 1991.

[12] J. C. Platt, "Fast Training of Support Vector Machines using Sequential Minimal Optimization," *Advances in Kernel Methods*, pp. 185–208, 1999. [Online]. Available: http://portal.acm.org/citation.cfm?id=299094.299105