Detection of Harassment on Web 2.0

Dawei Yin[†]
day207@cse.lehigh.edu

Zhenzhen Xue[†]
zhx208@cse.lehigh.edu

Liangjie Hong[†]
lih307@cse.lehigh.edu

Brian D. Davison[†]
davison@cse.lehigh.edu

April Kontostathis[‡]
akontostathis@ursinus.edu

Lynne Edwards[§]
ledwards@ursinus.edu

[†]Department of Computer Science and Engineering
Lehigh University
Bethlehem, PA 18015 USA

[‡]Department of Mathematics and Computer Science
Ursinus College
Collegeville, PA 19426 USA

[§]Department of Media and Communication Studies
Ursinus College
Collegeville, PA 19426 USA

ABSTRACT

Web 2.0 has led to the development and evolution of web-based communities and applications. These communities provide places for information sharing and collaboration. They also open the door for inappropriate online activities, such as harassment, in which some users post messages in a virtual community that are intentionally offensive to other members of the community. It is a new and challenging task to detect online harassment; currently few systems attempt to solve this problem.

In this paper, we use a supervised learning approach for detecting harassment. Our technique employs content features, sentiment features, and contextual features of documents. The experimental results described herein show that our method achieves significant improvements over several baselines, including Term Frequency-Inverse Document Frequency (TFIDF) approaches. Identification of online harassment is feasible when TFIDF is supplemented with sentiment and contextual feature attributes.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval; I.7.5 [**Document and Text Processing**]: Document Analysis

General Terms

Algorithms, Experiment, Measurement

Keywords

Harassment, Misbehavior, Machine learning, TFIDF, SVM

1. INTRODUCTION

In Web 2.0, users are relatively free to publish almost anything in online communities. Some users take advantage of this openness at times by harassing others in a variety of ways. For example, a user might respond to suggestions or statement of opinion with foul language. Another user might clog a channel with long lines of gibberish. On the flip side, a user might become infatuated with someone in an online setting, and may engage in cyberstalking behavior. Some of these forms of harassment are direct extensions of classic human rudeness in interpersonal relationships; others can only be found online. It is widely assumed that that anonymity in online communities can provide a 'safer' environment for the harasser, since her true identity remains hidden.

CAW2.0 2009, April 21, 2009, Madrid, Spain.

There are two basic types of virtual communities: discussion forums and chat rooms contained within a larger application (such as an online game). Both types of communities harbor harassers who prevent users from communicating and collaborating in a favorable way. As a result, most communities rely on user driven moderation systems, which require both time and human resources, to regulate user behavior.

Although harassment is a common problem in online communities, there are no existing approaches to detect online harassment automatically and effectively. This is likely the result of a number of issues. For example, while methods effective in detecting spam in social media (e.g., [14, 11, 13, 15]) provide some guidance, most of these approaches are not suitable to the problem of detecting harassment. This task is also sufficiently narrow that no standard, labeled dataset has been available for study. Importantly, our examination has found that the ratio of harassment documents to normal documents is small, and thus collecting enough training data for model development and training is a challenging subtask that must be overcome during the development of a harassment detection system. A lack of samples makes it hard to identify all the features and attributes which characterize harassment. Furthermore, many humans find the task of labeling harassment to be difficult in part because of inherent ambiguity. For example, good friends may use sarcasm in their discussion, and that sarcasm could be perceived by an observer as harassment. The potential for different interpretations of a word or phrase, especially if it is taken out of context or if the discussion participants have a long relationship with each other increases the difficulty in creating a standard dataset.

In this work, we define harassment intuitively as communication in which a user intentionally annoys one or more others in a web community. We address the detection of harassment as a classification problem with two classes: positive class for documents which contain harassment and negative class for documents which do not contain harassment. In our case, we treat each post as a document. Each document either belongs to the positive class or the negative class. We make use of a variety of attributes, including local features, sentiment features, and context features, to classify each document into one of the two classes. Our experiments show that a harassment detection classifier can be developed. We regard these early results as a good starting point for further research in this area and for related problems, such as identification of malicious users and detection of online criminal activities.

Our work has two primary contributions. First, we provide a reasonable definition for online harassment with a focus on detecting intentional annoyance. Second, we propose a supervised learning approach for harassment detection, and show its effectiveness using

three different test datasets.

The remainder of this paper is organized as follows: Section 2 surveys related work. In Section 3 we define key concepts which are used through the paper. Section 4 provides an overview of our proposed harassment detection method. An evaluation of our method applied to three typical Web 2.0 datasets is presented in Section 5. Finally, Section 6 concludes the paper and describes avenues for future work.

2. RELATED WORK

Online harassment is but one kind of online undesirable content. Elsewhere [12] we describe preliminary work toward the tracking and categorization of Internet predators. While material generated by predators is certainly undesirable to society, it is intended to be welcome to the (would be) victims, and thus not the type investigated here.

In this section, we will discuss research efforts that on similar tasks in two research areas: social media spam detection, and opinion mining or sentiment analysis.

2.1 Social Media Spam Detection

Spamming on social media, especially comment spam and forum spam, is a type of spam that prevents normal interactions among users. Typically, comment spam and forum spam violate current context because they pertain to completely different issues and topics. Users may find this kind of spam annoying, especially when it takes forms such as consecutive replies or long lists of keywords and advertisements, and it is, therefore, related to our definition of online harassment. The key difference between spamming and harassment is that harassment may or may not follow existing discussion topics while spamming is usually off-topic. In addition, unlike spamming, online harassment generally does not have potential commercial purpose, such as promotion of affiliated web sites.

Recently, spam detection on social media has attracted extensive research attention. Mishne et al. [14] used language model disagreement to detect blog comment spam and showed promising results. Their basic assumption is that the spam comments are off-topic comments, and therefore they may have different language models than normal comments. Kolari et al. [11] proposed a machine learning approach for blog spam detection that utilized content features, anchor text features, URL link features and several special features. We employ some similar features in our approach, but the lack of anchor text and URL links makes our task more difficult. Lin et al. [13] used temporal dynamics and self-similarity analysis for blog spam detection. This kind of information is usually not available on social media. Niu et al. [15] did a quantitative study on forum spam and used content-based features to identify several different types of forum spam.

We note that most methods used in spam detection on social media cannot be directly applied to harassment detection because we do not have access to the same features, such as links and anchor text.

2.2 Opinion Mining

The opinion mining task aims to extract opinions and reviews from user generated content. This area is also known as sentiment analysis or sentiment classification. One of the tasks in opinion mining is determining whether users hold positive, negative or neutral attitudes towards certain entities (e.g., products or features). This activity is similar to our task, which aims to identify depreciating remarks. Extensive work [5, 4, 7, 6] has been done in mining user opinions on products.

Recently, researchers have started to focus on review spam and untrusted review detection because it contains false positive or malicious negative opinions. Jindal and Liu [9, 8, 10] used textual features and applied logistic regression to identify review spam. Their research gave an average AUC value of 78%. Although harassment detection task shares the goal of finding negative or depreciating remarks, harassment detection focuses on a broader array of topics and expressions, and therefore we cannot rely only on local textual features.

3. DEFINITIONS

In this section, we introduce some key concepts and terms which are used through the paper.

As a starting point, we need a definition for online ***harassment***. Since harassment is a relatively general and fuzzy term, we define it restrictively as a kind of action in which a user intentionally annoys one or more other users in a web community. In this work, we focus on a specific kind of harassment in which a user systematically deprecates the contributions of another user. There are some other kinds of harassment which could be consider. For example, a person may be too keen to establish connections with another user who is not interested in these connections is also considered as harassing to others. However, because of the limitations of our experimental datasets which contain few examples for other kinds of harassment, we mainly focus on the personal insult harassment as defined above.

For example, the following excerpts show three examples of harassment from the datasets in our experiments (the first two are from Slashdot and the last one is from MySpace). The first example is quite obvious, a harassment which shows explicit rudeness. The second example of harassment is not as apparent as the first one since the choice of words is more "polite". The third one is an example of non-harassment, which although it uses foul language and appears to be rude, it is not meant to intentionally annoy others. The examples remain exactly the same as they are in the original datasets, including spelling errors and abbreviations.

1. Of all the stupid things you've said, this is by far, the most fucking stupid thing. You're really an idiot, you know that?

2. Thank you for showing all of us your own obvious lack of education. Not to mention culture.

3. so can u explain his platform? cuz i cant figure that fuckin shit out. dude blows a lot of smoke up my ass, but i don't see much substance there.

In our paper, we mainly focus on two kinds of communities, ***discussion-style*** and ***chat-style***. In discussion-style environments, there are various ***threads***, usually with multiple ***posts*** in each thread. Users are free to start a new thread or participate in an existing thread by adding posts to it. Usually, the discussion within one thread pertains to one predefined topic.

In chat-style communities, the conversations are more casual, and each post usually consists of only several words. Most of the time, there is little very little information about the existence (or absence) of a main topic in such conversations.

We used three experimental datasets; one is representative of chat-style communities (e.g., Kongregate) while the other two are discussion-style communities (e.g., MySpace). Each dataset consists of many threads, and each thread has multiple posts. Information available for each post includes the author of the post, the content of the post, which is also referred to as the ***post body*** and the time when the post is published. In our experiments, we judged

a post as harassing or not based on the post body. A ***positive post*** is a post which is considering harassing according to our definition. So, the input to our method is a post and the output is a judgment for the post, either positive or negative.

4. A MODEL FOR ONLINE HARASSMENT DETECTION

We now describe our proposed method for detecting online harassment. As mentioned earlier, our method addresses the problem using a supervised learning approach. We train a classifier with manually labeled posts and their corresponding features. Once a model is developed based on this training data, the model is used to classify each post in the test data set as either positive (a post contains harassment) or negative (a post does not contain harassment).

We use three kinds of features, namely, local features, sentiment features and contextual features. We now describe each of these feature types.

4.1 Local Features

The most basic features that we select into the model are local features, that is, features which can be extracted from a post itself. We use each distinct term as one feature and calculate a Term Frequency/Inverse Document Frequency (TFIDF) value for each feature. The TFIDF weight for term i in post j is:

$$TFIDF_{ij} = TF_{ij} \cdot IDF_i$$

Term frequency provides a measure of how important a particular term is in a given post (a local weighting). It is defined as:

$$TF_{ij} = \frac{n_{ij}}{\sum_k n_{kj}}$$

where n_{ij} is the number of occurrences of term i in post j, and the denominator is the count of the occurrences of all terms in post j. TF_{ij} will be larger for terms that appear more often in a post.

The inverse document frequency provides a measure of how important a particular term is within the entire corpus (a global weighting). IDF for term i is defined as:

$$IDF_i = \log \frac{|P|}{|\{p_j : t_i \in p_j\}|}$$

where $|P|$ is the total number of posts in a dataset, $|\{p_j : t_i \in p_j\}|$ is the number of posts in which term t_i appears. IDF scores are higher for terms which are good discriminators between posts (i.e., terms appearing in many posts will receive lower IDF scores).

Thus, each post is represented as a vector of terms and each term is represented in the vector by its TFIDF value. Terms that appear in the corpus but not in a given post will receive a TFIDF weight of 0 (because TF will be 0).

4.2 Sentiment Features

We use more specific features for specific detection of harassment.

After a careful review of the test data sets, we noted that many harassment posts contain foul language. The authors of harassment posts usually use offensive words to attack others in the community. So, foul language may be regarded as a sign of harassment. We also noticed that when people are harassing others, they tend to use personal pronouns. Thus, a personal pronoun appearing near profanity is a good indicator of harassment. In addition, we have observed that the second person pronouns such as "you" and "yourself" play a more important role among all possible pronouns. Table 1 shows some sample harassment formats.

1	BadWord!, Pronoun...
2	You BadWord...
3	I BadWord Pronoun ...
4	Pronoun BadWord ...
5	Pronoun ... BadWord ...

Table 1: Patterns of harassment.

Our method captures the following sentiment features for use in the development of our model:

1. Second person pronouns. For this feature, we treat all the second person pronouns in a post, including "you", "your" and "yourself", as a single term, and calculate the TFIDF value for this term for each post. Grouping all second person pronouns into one term reinforces the effect. Although this feature is similar to the next feature, we separated it because we believe that second person pronouns are more important than other personal pronouns for harassment detection.

2. All other pronouns. Similar to the first feature, we group all other pronouns and represent their weight using the TFIDF function. Second person pronouns are excluded, and therefore this grouping includes "he", "his", "himself", "she", "her", and "herself", as a single term.

3. Foul language. In this feature, we treat all the profanity in our dictionary as one single term, and then calculate the TFIDF value for this term for each post. Since foul language appears sparsely in the dataset, the grouping strategy is again used to reinforce the effect of this feature.

4.3 Contextual Features

Using the features extracted from a post itself to detect harassment is insufficient. When manually labeling training data, we often needed to look at the context of a post to make a decision.

In virtual communities, posts with many personal pronouns and foul language are sometimes not harassment. Users who have strong opinions on a topic, or who are embroiled in a debate, tend to use phrases or words which make the post look like harassment. Other scenarios, such as users who are familiar with each other and communicating in a very casual and informal way, can also appear to be harassing when they are considered alone.

So, we identified other features which can distinguish such harassment-like posts from real harassment posts. We refer to these features as contextual features. The vast majority of posts in a corpus are non-harassment, and a harassment post appears different from its neighboring posts (the posts which surround the target post). So, posts which are dramatically different from their neighbors are more likely to be harassment. However, when the first harassing post appears, it will often cause other users to respond with retaliatory harassment. A group of harassing posts will form a cluster. The contextual features, defined below, aim to find such relationships among posts.

1. Similarity feature. Defined as follows:

$$\sum_{p' \in \{N(k,p), P(1), Avg(P)\}} Sim(p, p')$$

where p is a post to be evaluated, $N(k, p)$ is a set of all neighbor posts of p, $P(1)$ is the first post in a thread, $Avg(P)$ is the average information of all the posts within a thread, and $Sim(p, p')$ is a function, specifically cosine similarity function in our system, to calculate the similarity between post p

and its neighbor post p'. A neighbor post is either one of the most recent k posts before post p, or is one of the k succeeding posts after post p. In our system, each post is represented as its TFIDF vector. So, the average information of a thread $Avg(P)$ is represented as a vector which averages all its post vectors in a thread. In addition to comparing each post with its k neighbors, we also compare it with the first post of a thread. This is based on the heuristic that the first post defines the topic of the thread (this heuristic usually holds in discussion-style communities, but it not always in chat-style communities). Similarly, we also compare a post with the average information of a thread because most posts are related to the same topic. Posts which are different from the thread average have the potential to be harassment.

2. Contextual post feature. It is defined as follows:

$$\sum_{p' \in N(k,p)} p'$$

where p is a post to be evaluated, $N(k, p)$ is a set of all neighbor posts of p, where neighbor posts are defined in the same way as in similarity feature. In this feature, each post is represented as the sum of its neighboring posts (vector sum). This feature was defined based on the assumption there will be a reaction in posts which are near a harassment post. So, the cluster of posts near a harassment post should look different from the cluster of posts which are near normal posts.

5. EXPERIMENTS

In this section, we describe the details of datasets and experiments. We compare our approach with three basic methods and show its effectiveness.

5.1 Experimental Setup

We employ libSVM [2] with the linear kernel as our classification tool. For ease of experimental replication and to avoid overfitting, all tool parameters were set to their default values. Cross validation was used to compare the performance of different classifiers. In K-fold cross-validation, the original sample is randomly partitioned into K subsamples. Of the K subsamples, a single subsample is retained for testing the model, and the remaining $K-1$ subsamples are used as training data. The cross-validation process is then repeated K times (the folds), with each of the K subsamples used exactly once as the validation data. The K results from the folds then are averaged to produce a single estimation. In our experiment, $K = 10$.

Before extracting features, we pre-processed the data. We tokenized each post and stemmed the words. Because the ratio of positive posts to negative posts are very small (roughly 1:77), a default SVM achieved optimal performance by merely identifying all samples as non-harassing. Therefore, we replicated the positive samples in order to train the classifier to detect harassment. All of our experiments employed the replication of positive instances strategy, except as noted below. After replication the ratio was changed to:

$$\frac{Number of Positive Posts}{Number of Negative Posts} = 1/2$$

We use the following metrics to evaluate the experimental results. **Precision**: the percent of identified posts that are truly harassment. **Recall**: the percent of harassment posts that are correctly identified. **F-measure**: the weighted harmonic mean of precision and recall. We used F_1, which gives equal weight to precision and recall, and which is defined as:

$$F_1 = \frac{2 \cdot (Precision \cdot Recall)}{Precision + Recall}$$

	Kongregate	**Slashdot**	**MySpace**
number of threads	1	17	148
number of posts	4,802	4,303	1,946
positive posts	42	60	65

Table 2: Size of each labeled dataset.

5.2 Datasets

Fundación Barcelona Media (FBM) provided five datasets for analysis in the CAW 2.0 workshop [3]. Our experiments make use of three of them: Kongregate, Slashdot and MySpace. Kongregate is a web site which primarily provides online games to be played using a web browser with the Adobe Flash plugin. It also provides chat-rooms for real-time communication among players. Slashdot is a popular website for people who are interested in reading about and discussing technology and its ramifications. MySpace is a popular social networking site which offers its registered users the opportunity to participate in forum discussions about predefined topics.

These three datasets can be grouped into the two kinds of communities as we introduced in Section 3. Kongregate is a chat-style community, and Slashdot and MySpace are discussion-style communities. In chat-style communities like Kongregate, posts are usually short online messages which contain only a few words with many misspellings. In discussion-style communities, like Slashdot and MySpace, posts are relatively longer (but still shorter than full web pages) and the usage of terms in these posts is more formal, as compared with that of chat-style communities.

We manually labeled a randomly selected subset of the threads from each labeled dataset. The size of each labeled dataset is shown in Table 2. There is only one thread in Kongregate data set, and this thread contains 4,802 posts. Most of the posts are quite short, with, on average, just 5.3 terms per post. We labeled 42 as positive samples (harassment). In the Slashdot dataset, we read through 17 threads which include 4,303 posts in total and labeled 60 posts as positive. The MySpace dataset has a relatively high ratio of harassment, as compared to Slashdot and Kongregate. A total of 65 positive posts were found among 148 threads (containing a total of 1,946 posts).

5.3 Experimental Results

During the development of our harassment detection model, we tried three basic approaches. We compare harassment detection algorithms to these approaches in Section 5.3.4.

5.3.1 N-gram

We first used word-level N-grams as binary features. This approach has been used in many information retrieval tasks, such as spam detection, and sentiment identification, and has been success for those activities. In our experiment, 1-gram, 2-gram and 3-gram were used. In order to keep the dimensions to a reasonable scale, we filter out grams with relatively low frequencies, leaving use with approximately 5,000 N-grams (1-grams, 2-grams, and 3-grams combined). The statistical results using N-gram as binary features are shown in Table 3.

Use of N-gram to detect harassment is not very effective. All three datasets had F_1 statistics around 15%. The recall for MySpace was a little better, but the precision was still a very low 11%.

	Kongregate	Slashdot	MySpace
Precision	0.139	0.179	0.110
Recall	0.140	0.117	0.354
F-measure	0.140	0.141	0.168

Table 3: Performance using N-grams.

We believe there are several reasons for the poor performance of N-grams for harassment detection. First, in some harassment posts, infrequent words, which could make the posts distinct from normal posts, are used. Those words may have been filtered out in our pruning during feature extraction. Secondly, we used binary representations for our n-grams, not local and global weighting formulas such as TFIDF. Finally, in our manual analysis identified important information from neighboring posts, but N-gram features only contain local information.

5.3.2 Foul Language

Most of the harassment posts contain profanity. Although the occurrence of foul language is not an absolute sign of harassment, it can be an important clue toward harassment detection. We tried a simple experiment which used only foul language for harassment detection. We downloaded a dictionary of profanity from [1], but did not add special weights to any single word. Our strategy used each profane term as a one feature. The value of a feature equals the number of occurrences of the term in a post. As we expected, in most posts, the feature vector is very sparse. On average, as few as two to four features are nonzero. Table 4 shows the results of matching terms considered foul language.

As we can see from Table 4, this method was also not very effective. All F-measures are below 20%. The result shows that the use of only foul language is insufficient for harassment detection. Spelling errors occur frequently in online chat and, therefore, some of the profanity used in the chat logs were not in the dictionary we used. Also, the foul language words are not absolute signals. Some harassment uses a euphemistic style which does not use foul language. On the converse side, teenagers sometimes use some bad words to express strong emotion, but this does not necessarily indicate harassment. Although foul language features do not work very well when applied alone, as we will see below they can be an important component for identifying harassment when used as part of a larger model.

5.3.3 TFIDF Features

Another basic method which has been used in spam detection is terms with TFIDF weighting (also used as the local feature in our model). In this experiment, we only extract terms with a frequency greater than 1 in the corpus, to filter out useless terms and also to reduce the dimensionality of the feature space.

Often TFIDF weighting is no better than N-gram. N-grams with N $>$ 1 can capture information about sequences of terms, while TFIDF only incorporates the effect of individual terms. However, there are several arguments in favor of TFIDF for our task. First, TFIDF quantifies the weight of each term in the post. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. To some extent, these TFIDF weights can define sensitive words which are frequently used in harassment. TFIDF weighting is usually applied to 1-grams, a fraction of all n-grams, eliminating the need to filter features and potentially lose information. In addition, in the discussion-style posts, the language used has fewer spelling errors. The words which are frequently used in harassment will receive higher weights, and these discriminating features can be identified by the classifier. Since harassment posts are usually very short (longer posts are more likely to be arguments, even if they contain foul language or strong emotion), and TFIDF weighting takes post length into consideration. The results using TFIDF weighting only appear in Table 5.

	Kongregate	Slashdot	MySpace
Precision	0.289	0.273	0.351
Recall	0.571	0.231	0.217
F-measure	0.384	0.250	0.268

Table 5: Performance of TFIDF-weighted features.

The results are better than both the N-gram and foul language approaches. All of the metrics are above 20% and the best one, recall for the Kongregate dataset, reaches 57%. TFIDF is much more effective than other basic methods for detecting harassment. However, the performance of TFIDF is still far from our expectation. We designed our model to improve upon the TFIDF performance.

5.3.4 Combining TFIDF with Sentiment and Contextual Features

The low performance of simple methods indicates that more sophisticated methods are required to detect harassment. Our experimental approach involves use of local features, sentiment features and contextual features. For the contextual features, we set the window size parameter to $k = 3$. For the sentiment features, the dictionary is the same as the profanity dictionary used in our foul language experiments.

Table 6 shows the retrieval results when our model was used for harassment detection. The results are much better than the simpler methods described above; all performance metrics are at or above 25%. The recall of Kongregate is near 60%. The F-measure for both Slashdot and MySpace are near 30%. The precision of MySpace is above 40%.

1. **Comparison with the TFIDF experiments**

   Comparing the experimental model to the TFIDF-only model, we can see the combined features improve the performance in Kongregate by about 6% on F-measure, and on the discussion-style dataset such as Slashdot and MySpace, the F-measure increases by approximately 5%.

   In both chat-style and discussion-style communities, harassment posts often appear in clusters. The contextual features are helpful in detecting this harassment.

	Kongregate	Slashdot	MySpace
Precision	0.500	0.104	0.154
Recall	0.095	0.200	0.031
F-measure	0.160	0.137	0.051

Table 4: Performance of matching foul language.

	Kongregate	Slashdot	MySpace
Precision	0.352	0.321	0.417
Recall	0.595	0.277	0.250
F-measure	0.442	0.298	0.313

Table 6: Performance of our combined model.

	Kongregate
Precision	0.394
Recall	0.619
F-measure	0.481

Table 7: TFIDF and contextual features.

2. **Comparison between datasets**

   The data in Table 6 suggests that combined features provide a greater benefit in chat-style communities as opposed to discussion-style communities, although detection of harassment improves in both types of communities when contextual and sentiment features are added to the basic TFIDF system. This is because chat-style content is characterized by very short posts (often only a few words). If we only use TFIDF, the amount of useful information is very limited. In chat-style discussion, harassment posts are often off topic, and thus can be detected effectively by our similarity features.

   F-measure improvement in discussion-style forums is more muted because we only use the time sequence to identify neighbors for our contextual features. In discussion-style communities, posts within a thread are about a common topic and users often reply to previous posts. In this case, a previous post is replied by another user in a later post, and these two posts are the real neighbors to each other. These two posts may not be adjacent and there may be several or many posts between them in the time sequence. Here, if a post is published in reply to a previous post, they are defined to be in the same branch. Discussion-type communities often contain multiple branches within a single thread. Unfortunately, meta-data about branching is not available in the MySpace or Slashdot data. In future work, we may extract new datasets which have branching information to see if we can find attributes that provide better contextual information.

3. **Further improvement**

   Spelling errors occur frequently in the chat-room dataset, but less frequently in the discussion-style dataset. If there are too many spelling errors on chat-style content, and almost no complete sentences, the sentiment features may not work on the chat-style dataset. In some cases, the sentiment features can become noise in our system. Table 7 verifies our conjecture. When we only use contextual features and TFIDF feature, we get even better results on the chat-style dataset (Kongregate), but performance is worse for the discussion-style data sets, MySpace and Slashdot.

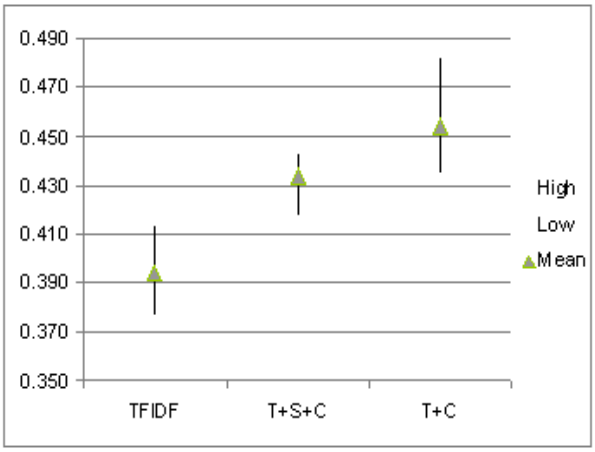

Figure 1: Mean F_1 performance on the Kongregate dataset.

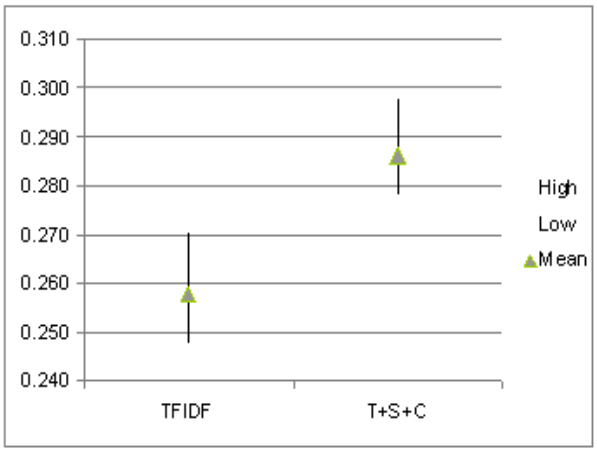

Figure 2: Mean F_1 performance on the MySpace dataset.

Because our experimental process randomly partitions the original sample into ten subsamples, it is possible for the partitioning to affect the final result. In order to eliminate the effect of randomization, we perform ten-fold cross validation five times. Each time the original sample is randomly partitioned, and the ten subsamples are different from each other. Figures 1, 2, and 3 show the results of our experiments for each of the data sets. S refers to the sentiment features, T stands for the terms with TFIDF weighting, and C refers to the contextual features. The results in figures show that our method stably improves the performance, as compared to the TFIDF only experiments.

6. CONCLUSIONS AND FUTURE WORK

Harassment communication is unique and harassment detection requires the development of new methods and features. We have presented some initial methods for identifying harassment using supervised learning. Our final approach combines local features,

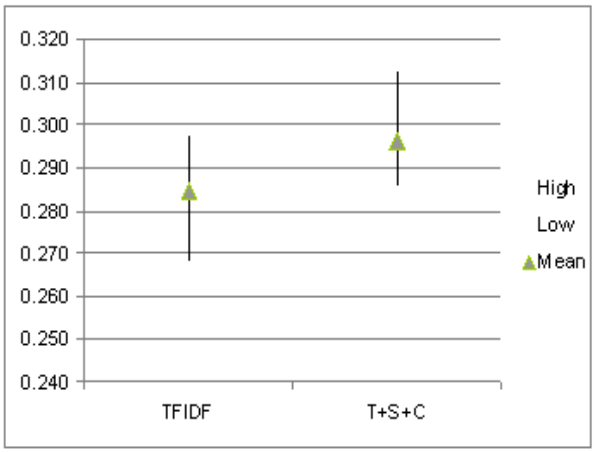

Figure 3: Mean F_1 performance on the Slashdot dataset.

sentiment features, and contextual features to train a model for detecting harassing posts in chat rooms and discussion forums. Our experiments show that the addition of the sentiment and contextual features provide significantly improved performance to a basic TFIDF model.

Our system introduces contextual and similarity features. Future work will result in a refinement of these ideas and metrics. Future research may also involve other features; we have not yet fully utilized temporal or user information. In our experiments to date, only supervised methods are employed; unsupervised methods may also prove valuable.

7. REFERENCES

[1] http://www.noswearing.com/dictionary.

[2] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[3] Fundación Barcelona Media (FBM). Caw 2.0 training datasets. Available from http://caw2.barcelonamedia.org/, 2009.

[4] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 168–177, New York, NY, 2004. ACM Press.

[5] M. Hu and B. Liu. Mining opinion features in customer reviews. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI)*, pages 755–760, 2004.

[6] N. Jindal and B. Liu. Identifying comparative sentences in text documents. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 244–251, New York, NY, July 2006. ACM Press.

[7] N. Jindal and B. Liu. Mining comparative sentences and relations. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, 2006.

[8] N. Jindal and B. Liu. Analyzing and detecting review spam. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM)*, pages 547–552, Los Alamitos, CA, 2007. IEEE Computer Society.

[9] N. Jindal and B. Liu. Review spam detection. In *Proceedings of the 16th International Conference on the World Wide Web (WWW)*, pages 1189–1190, New York, NY, May 2007. ACM Press.

[10] N. Jindal and B. Liu. Opinion spam and analysis. In *Proceedings of the International Conference on Web Search and Web Data Mining (WSDM)*, pages 219–230, New York, NY, 2008. ACM Press.

[11] P. Kolari, A. Java, T. Finin, T. Oates, and A. Joshi. Detecting spam blogs: A machine learning approach. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, July 2006.

[12] A. Kontostathis, L. Edwards, and A. Leatherman. ChatCoder: Toward the tracking and categorization of internet predators. In *Proceedings of the 7th Text Mining Workshop*, May 2009. Held in conjunction with the 9th SIAM International Conference on Data Mining (SDM). To be published.

[13] Y.-R. Lin, H. Sundaram, Y. Chi, J. Tatemura, and B. L. Tseng. Detecting splogs via temporal dynamics using self-similarity analysis. *ACM Trans. Web*, 2(1):1–35, 2008.

[14] G. Mishne, D. Carmel, and R. Lempel. Blocking blog spam with language model disagreement. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, May 2005.

[15] Y. Niu, Y. min Wang, H. Chen, M. Ma, and F. Hsu. A quantitative study of forum spamming using context-based analysis. In *Proceedings of the 14th Annual Network and Distributed System Security Symposium (NDSS)*, 2007.