Automatic identification of personal insults on social news sites

Sara Owsley Sood
Pomona College
185 East Sixth Street
Claremont, CA 91711
sara.owsleysood@pomona.edu

Elizabeth F. Churchill, Judd Antin
Yahoo! Research
4301 Great America Parkway
Santa Clara, CA 95054
<echu, jantin>@yahoo-inc.com

Abstract
As online communities grow and the volume of user-generated content increases, the need for community management also rises. Community management has three main purposes: to create a positive experience for existing participants, to promote appropriate, socio-normative behaviors, and to encourage potential participants to make contributions. Research indicates that the quality of content a potential participant sees on a site is highly influential; off-topic, negative comments with malicious intent are a particularly strong boundary to participation or set the tone for encouraging similar contributions. A problem for community managers, therefore, is the detection and elimination of such undesirable content. As a community grows this undertaking becomes more daunting. Can an automated system aid community managers in this task? In this paper, we address this question through a machine learning approach to automatic detection of inappropriate negative user contributions. Our training corpus is a set of comments from a news commenting site that we tasked Amazon Mechanical Turk workers with labeling. Each comment is labeled for the presence of profanity, insults, and the object of the insults. Support vector machines trained on this data are combined with relevance and valence analysis systems in a multistep approach to the detection of inappropriate negative user contributions. The system shows great potential for semi-automated community management.

Keywords
online communities, sentiment analysis, comment threads, user-generated content, emotion, negativity, community management

1. Introduction
User-generated content is fundamental to the notion of the "social web." However, an everyday issue facing technology companies and application developers is the encouragement of active user participation. One barrier to participation is the presence of negative content – messages that are not merely contentious, but angry, hostile, or abusive. If such messages are out of line with the general tenor of the site, this content can drive existing participants away and deter newcomers. For those who do participate, the presence of negative content signals that aggressive or abusive contributions are tolerated or are perhaps even normative (Sukumaran, Vezich, McHugh, & Nass, 2011). Studies of 'griefers' and 'trolls' (people who engage in transgressive behaviors such as posting general negative meta-comments and targeted "bully" like insults as a form of sport) suggest these behaviors are a serious problem (Chesney, Coyne, Logan, & Madden, 2009).

While negative content on user-generated sites takes many different forms, in this article, we are concerned with identifying negative content that is offered with malicious intent. Distinguishing negative content offered in the spirit of conversation and debate from that with malicious intent is non-trivial, and often requires a great deal of nuance and contextual knowledge. Furthermore, norms about what types of negative content is considered inappropriate can vary according to the topic of discussion and the particular website. From this we believe it is worthwhile to distinguish *off-topic negative comments* from *on-topic negative comments* that, while negative, are offered in the spirit of debate.

There is an interesting tension between encouraging debate and moderating negativity; negative content may discourage participation, but a lack of negative content may also indicate lack of engaged, lively debate. Completely sanitized sites may not provide an interesting forum and thus may not attract repeat visitors. A central set of tasks for user-generated content sites, therefore, is to (1) clearly establish the latitude and goals of the site (2) establish and incent 'reasonable' and 'appropriate' behaviors while disincenting disruptive behaviors, (3) identify malicious content as it is produced, (4) eliminate it where appropriate, and (5) identify repeat offenders (Gazan, 2009; Lampe & Johnston, 2005; Lampe & Resnick, 2004).

For any site that relies on user-generated content, the need for clarity of the site's social contract with users grows as it becomes more popular. The scale of moderation and management also grows. Not all sites have good tools for nuanced moderation, even if they have excellent spam detection. Community managers, typically appointed and employed by the product group of a website, "eyeball" content for contributions that are at odds with the community guidelines and the tolerance levels of other contributors. Such "eyeballing" implies that community managers and moderators must scan sometimes vast amounts of content to locate inappropriate contributions and behaviors. Efforts towards 'social moderation' in which users flag others' comments as inappropriate have great potential but are largely unused on many sites (including the one studied in this article) and are vulnerable to collusion (Lou, Chen, & Lei, 2009). Thus, community managers play a very important role in moderating disputes and removing inappropriate content from the community, in addition to external outreach promoting the community/site through social media and other venues.

In this article, we address the possibility of an automated system to aid community managers in the task of identifying and removing malicious content. We investigate whether such a system is feasible by considering what type, quality and quantity of training data are needed. Knowing that words and terms are understood based on context of use (e.g., 'Falling off the wagon' is an idiomatic phrase in some cultures and would be interpreted very differently if uttered in a bar versus in a playground), we wish to understand the extent to which the task is domain or site specific, or whether a single system would suffice to reduce the scale of this problem for most sites – greatly reducing the number of comments that a community manager must "eyeball." Since different sites have different social contracts and normative behaviors, we are not proposing a system that replaces community managers, but rather a tool a community manager can use to reduce the number of comments that they must review.

We approach these questions using a training corpus of a set of comments from a news commenting site. We tasked Amazon Mechanical Turk workers with labeling the comments. Each comment is labeled for the presence/absence of profanity and insults; for insults, the object of the insult was also labeled. We hypothesize that these labels might help identify those malicious comments. Support vector machines trained on this data are combined with relevance and valence analysis systems in a multistep approach to identifying negative comments offered with malicious intent. We present evaluations of our systems and illustrate their potential for semi-automated community management.

2. Background & Prior Research

The approach presented in this article is informed by and related to past research in the space of sentiment analysis. Since 2002, many researchers have focused their efforts on the task of automatically analyzing the sentiment of a document (typically represented by valence - how positive/negative it is) as expressed by its author by analyzing words that are used in the text. Sentiment or opinion analysis applies to any stream of textual information, especially user-generated content, such as blogs, web pages, bulletin boards, emails, chat rooms, and so on. Much of this work began with, and is still utilizing movie and product reviews as labeled training data (where the associated star rating for a review serves as the label). Many have used this data in supervised machine learning systems (Pang & Lillian Lee, 2008; Pang, Lillian Lee, & Vaithyanathan, 2002) while others have taken an unsupervised approach to building such systems, leveraging relationships between the words in the target document in order to calculate an overall sentiment score (Turney, 2002).

Given that such systems exist and are quite accurate, some as high as 90% (Anthony Aue & Michael Gamon, 2005), it seems as though utilizing these systems on our data (user comments on news stories) is a trivial task. However, it is well known that sentiment analysis is, in addition to being author, context and community-specific, a domain-specific problem. That is, words used in a positive context to describe cars are not necessarily the same words used in a positive way to describe movies. In fact, many words have opposing emotional connotations across domains. For example, a 'cold' beverage is good while a 'cold' politician is bad (Owsley, Sanjay Sood, & Hammond, 2006). Further, a text may say that a policy is "not at all desirable" (negative sentiment), or a product is "terribly good" (positive sentiment); detecting the underlying sentiment behind these kinds of statements is hard if simple lexicons are used. This implies that, in order to build an accurate sentiment analysis system, you must have labeled training data from within the target domain. As sufficient data sets do not exist in all domains, many have made efforts toward building systems to customize sentiment analysis systems to new domains without a large amount of labeled training data in the target domain (Anthony Aue & Michael Gamon, 2005; Gamon & Aue, 2005; S. Sood, Owsley, Hammond, & Birnbaum, 2007).

Most work in sentiment analysis has focused on building systems that simply indicate whether a document is positive, negative or neutral. Some have built systems to judge valence on multipoint scales (Pang & Lillian Lee, 2005), but little work has moved beyond the dimension of valence (a measure of the author's sentiment toward a topic – how positive/negative they are). However, there are a number of more complex models of emotion or sentiment that have been used when classifying human behavior. The simplest of these is the VAD or PAD Model, which characterize emotion on three dimensions representative of valence (or pleasure), intensity and dominance (M. M. Bradley & P. J. Lang, 1999). The best known is Ekman's "six emotion" model, which lays out the following as the six basic emotions that human beings experience: happiness, sadness, anger, surprise, disgust, and fear (Ekman, 2004). Some past work has included moving beyond the valence dimension and classifying documents based on the general 'mood' of the author, using a dataset of blog posts labeled with the author's stated mood as training data (S. Sood & Vasserman, 2009).

The current task offers many new challenges with regard to sentiment analysis; we address and approach three of them. First, there is no sentiment labeled dataset of this type (user comments on news stories) available for this task. Second, the comments themselves are rather short and often conversational; quite different from the self-contained longer documents given in movie and product reviews. Third, this task necessitates moving beyond the positive/negative judgment toward distinguishing negative comments of malicious intent from negative comments offered in the spirit of debate.

Similar to our task at hand, a few recent studies have approached the problem of cyberbullying among teens (Dinakar, Reichart, & Lieberman, 2011). One other group has completed a preliminary study aimed to detect online harassment, which they defined as "communication in which a user intentionally annoys one or more others in a web community" (Yin et al., 2009). Their work focused on data from three sources: Kongregate, Slashdot, and MySpace. While Kongregate contains very short 'chat' like messages, Slashdot and MySpace contain comment streams more similar to the comments in our present study. Yin, et al. manually labeled comments in these datasets for the presence of harassment, resulting in a relatively small dataset. Through various combinations of features, Yin, et al. were able to achieve maximal f-measures of 0.298 for detection of harassment on Slashdot, and 0.313 on MySpace. As Slashdot and MySpace were deemed most similar to the site we study here, these values serve as benchmarks for comparison to our work (Yin et al., 2009).

Our approach combines relevance analyses for detecting off-topic comments with valence analysis methods for detecting negative comments. With the observation that comments of malicious intent typically contain insults, we also approach the task of automated insult detection. Beyond detecting insults, we make a distinction between those insults that are directed at another user in the forum/community (the author of another comment) and those directed at a third party - under the assumption that the former is more likely to be malicious. In the sections that follow, we present a characterization of our data, as well as describe and evaluate our approach.

3. Dataset: Comments on a news site

In this article we focus on the management of user-generated comments on a general news site. Our particular focus is a "social news" website. The users of social news sites (e.g. digg.com, reddit.com) contribute links to news stories of interest. Frequently these stories are segmented into topics such as politics, entertainment, and sports. Other users then vote on contributed news stories using, for example, a "thumbs up" or "thumbs down" modality. The most popular stories are more prominently displayed in the site's ranked list, thereby creating a community-curated list of links to popular and interesting news stories. Central to our study, social news sites generally allow users to comment on each user-contributed story, thereby providing a space for conversation around each posting (See Figure 1 for an example posting).

NEWS | POLITICS

Number One Reason Sarah Palin Is Smiling Today

Letterman's embarrassing midnight confession

By ROBERT A. GEORGE

Updated 4:34 PM EDT, Fri, Oct 2, 2009

PRINT SHARE BUZZ UP! 1 retweet submit to digg FACEBOOK

Sarah Palin must be smiling, and not just because her new book is already #1 on the bestseller list, before it's even been released.

The ex-Alaska governnor's erstwhile nemesis, David Letterman, just confessed that he's been the Horndog King of Late Night all these years, telling his audience he is the victim of an extrotion plot and admitting, "I have had sex with women who worked for me on this show."

Dave? *Our* Dave with the gap-toothed, boyish grin? The guy who always joked about never getting girls? The one who -- when announcing that he was going to be a father with his long-time girlfriend -- said that, "I know what you're thinking -- Dave's had sex?" To make things sound even creepier, he used almost the exact same joke on Thursday: "I know what you're saying. I'll be darned, Dave had sex."

Figure 1: Story posted on a social news site. The story focuses on Sarah Palin's response to David Letterman's public confession about having affairs. This story generated many emotional comments, with varying degrees of relevance to the story.

Our dataset consists of 1,655,131 individual comments from 168,973 comment threads collected from a medium-size social news site, *Yahoo! Buzz*, between March 2010 and May 2010. *Yahoo! Buzz* was an experimental social news and commenting site that is no longer active. Each thread in our dataset contains an average of 9.8 comments (a standard deviation of 57.4) and represents the stream of comments contributed on a single news story. The modal number of comments per thread is 1, with a median of 1. This information is summarized in Table 1 and shown in graph form in Figure 2. The comments themselves varied greatly in length (as shown in Figure 3). The mean comment length was 43.9 words (with a standard deviation of 40.8), the modal length was 9 words, and the median length was 31 words. During the data collection period 234,855 unique users made at least one comment. Figure 4 illustrates the distribution of comments among users. The average number of comments per user was 7.05 (with a standard deviation of 60.5). The mode and median of comments per user were both 1. With respect to the number of comments per thread, the number of comments per user, and the number of words per comment our data closely follows the power law distribution, which is so common on a broad array of user-generated websites (M. Faloutsos, P. Faloutsos, & C. Faloutsos, 1999).

	Mean	**Standard deviation**	**Median**	**Mode**
# comments per thread	9.8	57.4	1	1
# words per comment	43.9	40.8	31	9
# comments per user	7.05	60.5	1	1

Table 1: Statistics that describe the dataset used in this study.

As mentioned above and shown in Table 1, our dataset (spanning three months) has an average of 7.05 comments per user, with a mode and median of 1. Furthermore, 67.45% of the users (158,423 of 234,855) represented on this site only made one comment (the mode) in the three-month period. This tells us that a majority of users of this site simply post their opinion and leave the site; they do not engage in a discussion or debate about the content. This relatively small number of comments per user is one indication that many users of the site are not drawn into the community and do not return repeatedly to participate and interact. This apparent lack of community affiliation, which is indicated in sparse and sporadic contributions by many different users increases potential for a broader range of contribution types. As such, our efforts towards automated community management are predominantly aimed at this end of the spectrum – sites without a tight knit community that is willing and able to moderate and curate the high volume of comments that these sites generally collect.

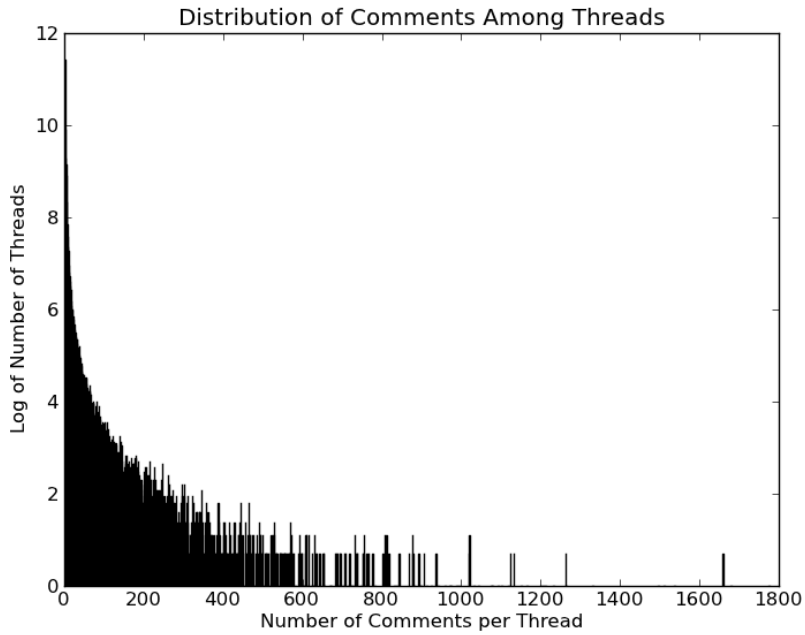

Figure 2: The distribution of comments among threads in the dataset.

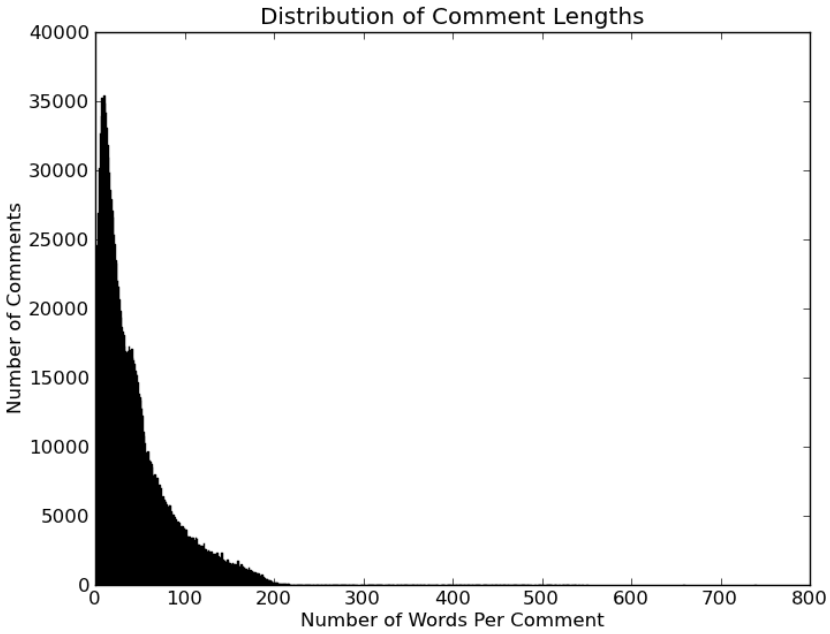

Figure 3: The distribution of comment lengths (in words) in the dataset.

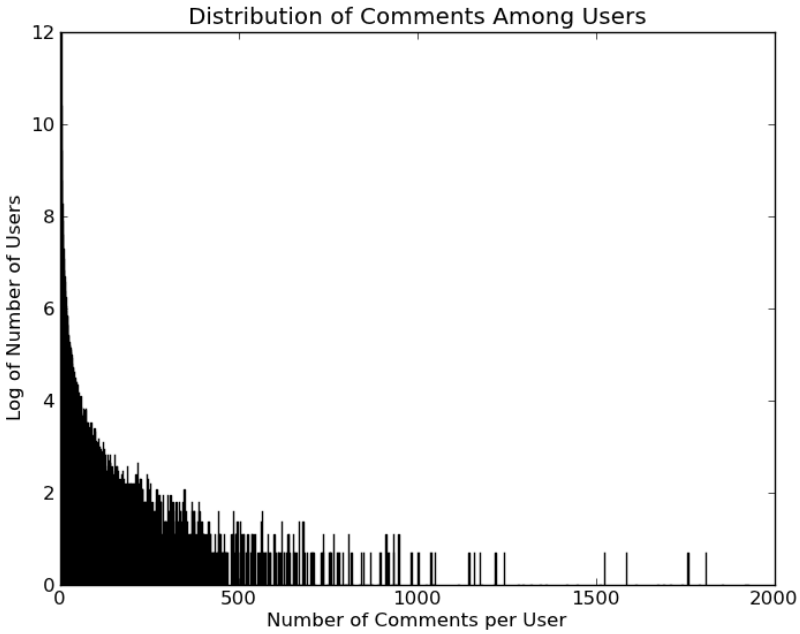

Figure 4: The distribution of comments among users in the dataset.

To illustrate the kinds of comments that are posted, below we offer an excerpt from a longer comment thread on a story from a Philadelphia-based online news site from October 2009[1]. The news item relates to a development in an ongoing public spat between Sarah Palin, the former running mate of the defeated Republican contender in the US Presidential Elections of 2008, John McCain, and David Letterman, a famous talk show host. In this particular story, Palin was reported as "smiling" at the revelation that David Letterman had admitted having extra-marital affairs. The

headline is shown in Figure 1. We selected this example because it illustrates several comment types and varying viewpoints.

C1[2]: At least he apologized; Palin accepted his apology so I'll follow her lead. She was right about Letterman, which we all knew and hopefully America realizes she has been right about OH so much more.
C2: Sarah Palin : "Told ya so!" GO SARAH !
C3: You go, girl, we're behind you 100%. Yes, that's right, blue dogs are Palin Democrats!
C4: The number 1 reason Sarah is smiling today, is because she just got a fat check for $7,000,000 bucks. Who wouldn't be smiling? Notice too, it's listed under the title, ENTERTAINMENT. That's the most truthful thing about this story. Obviously, she coughed up a substantial amount of the millions to get Todd to keep quiet about their impending divorce, by buying him a new boat and sending off into the Bering Sea to do some fishing. What he doesn't know yet is, the hull of the boat is made of bamboo and paper machete.
C5: The family that quits together splits apart.
C6: Man Palin is a has been stale news, good for small town hicks selling news to those like them. To them a pig farting is big news.
C7: STOP OFFENDING MY FELLOW AMERICANS.
C8: Say what you will. Palin rooked her naysayers. You can call her book whatever you can to dilute her success but, WHO'S LAUGHING NOW! The Clowns and Comics are presently in conference crying and consoling each other with their big towels wiping off their makeup and drying their big old shoes. Waaaaahhhhhh!
…
C9: I think the real reason Sarah is smiling is that so many people want to read her book! It is not only the money (everyone wants paid for their work) but the fact that Sarah can get the truth out to the Americans! Sarah does not like corruption or deceit! I want to know the truth too!
…
C10: Palin is a common whore. Let's dig of the past sex antics of Sarah and Toad Palin!
…
C11: The Palin family sucks, always have and always will. There are a few sex scandals in Sarah & Todd's past. Not to mention ethics problems etc. David is probably a nicer person than Sarah & Todd put together. She can smile at Dave's misfortune while conveniently forgetting her own sins.
…
C12: THE REASON SHES HAPPY!!!!!!!! WELL SHE TOOK HER FIRST SH^T IN MONTHS MMM MMM

The thread begins with three relatively positive comments in support of Sarah Palin, though the first is sympathetic to David Letterman. Peppered throughout the thread are various angry comments including comments both in support of Palin (e.g. C9) and angry conversational comments (e.g. C7). Though there is much variety in the sentiment conveyed in the posts, they clearly start out rather positive and become more negative over time. This is a pattern we observed across the entire dataset and concluded in a past study (S. O. Sood & Churchill, 2010). While different community managers could interpret this differently, some might consider comments such as C6, C10 and C12 to be malicious.

4. Coding Comments

While our dataset is rich in content – opinions, insults and debate – it is lacking in metadata about whether each comment would be deemed positive or negative. In order to build any meaningful community management system, we require information about what a human community manager might have to say about each comment or what action they might take. Which comments contain insults, profanity, are on-topic or off-topic, are extremely negative or positive? These are all questions that we seek to answer with an automated system, and thus must have labeled data in order to train and evaluate such a system.

4.1 The Pilot Coding Task

We conducted a pilot test in which the authors themselves labeled a set of comments. The primary goal of the pilot task was to iterate and refine the structure and design of the coding task, and test the questions and response scales that we would use for the primary coding task. The pilot revealed several issues, which helped improve the robustness of the coding task and the face validity of the response scales. First, the pilot quickly revealed that coding for positive or negative valence of a given comment was too difficult and vague to be useful for analysis. The valence response scale was dropped from the primary task. Secondly, we resolved a problematic confusion by rephrasing instructions so that coders would focus on the intent of the author when making judgments about insult or profanity. This change helped to make the task more specific – making

judgments about the intention of the author rather than simply whether anyone might label the text as insult or profanity. Finally, we clarified the task instructions by asking coders to consider both actual profanity and what we termed "disguised profanity." Disguised profanity refers to situations in which the comment's author clearly intended to convey a profane word or phrase, but disguised it through alternate spellings or the use of special characters.

4.2 Coding Comments with Mechanical Turk

Traditionally, to generate a coded dataset, researchers have employed a small number of dedicated content coders. In our study, however, we employed the crowdsourcing service Amazon Mechanical Turk (MTurk). MTurk is an online labor market in which requesters post jobs that can be easily decomposed into a large number of small tasks. MTurk workers ("Turkers") are presented with a short description of available tasks, and then choose which tasks to complete. Individual tasks typically take between 5 and 20 seconds to complete and workers are generally paid about 5 cents for each task. In addition to its function as a ready source of labor, MTurk has been noted as a useful venue for conducting user studies in human-computer interaction (Kittur, Chi, & Suh, 2008) as well as large-scale economic experiments (Mason & Watts, 2009).
MTurk is a relatively new channel for completing textual analysis, but it provides a number of significant benefits over the traditional model. First, Turkers are an on-demand labor force. Using the MTurk system, we were able to quickly and simultaneously engage a large number of coders, without the significant overhead associated with hiring dedicated workers. Secondly, MTurk workers come from many different perspectives and bring a variety of experiences with online interaction and comments. Our coders, for example, hailed from 36 different US states. Such geographical differences also likely translate to a valuable diversity of perspective in judgments about insulting or profane comments.

The quality of work provided by Turkers is obviously an important factor. Tetreault and colleagues showed that using MTurk for a similar content analysis task was both faster and more economical than using dedicated raters (Tetreault, Filatova, & Chodorow, 2010). Callison-Burch also found that employing multiple non-expert workers for a translation task could produce high quality results (Callison-Burch, 2009). Finally, Sheng and colleagues suggested that it is possible to achieve reliable, high quality coding by using multiple non-expert coders, even when those coders do not always agree (i.e. the data is "noisy") (Sheng, Provost, & Ipeirotis, 2008). While each of these tasks differs slightly from ours, this body of research suggests that MTurk is not only appropriate for content coding and textual analysis, but that it can efficiently produce high quality results.

A second key factor concerns the need to train workers to understand coding tasks. MTurk addresses this issue through the use of "gold" questions. Gold questions are training questions for which the correct answer has been predetermined. As new Turkers begin work, gold questions are mixed in with non-gold questions. When Turkers provide erroneous responses to gold questions, they are provided with the correct answer and an explanation. A worker who answers too many gold questions incorrectly is barred from completing further work and his responses are removed from the dataset. In effect, then, gold questions constitute a basic training program for Turkers.

The MTurk Dataset

To select a sample for coding via MTurk, we first filtered the complete set of 1,655,131 comments to include only those comments in the 2nd and 3rd quartiles for overall comment length (between 73 and 324 characters long). This was done in order to eliminate comments that were either too short to meaningfully interpret or too long to digest quickly. In addition we selected only English language comments contributed in threads that made reference to US news content. From this filtered set, we then randomly selected 6500 comments as the MTurk dataset. Over the course of approximately 5 days, 221 MTurk workers provided 25,965 judgments on 6500 comments.

Traditional measures of inter-rater reliability such as Fleiss Kappa require that a fixed set of coders provide responses for a fixed set of items. In our case, however, 221 different Turkers provided judgments. The most active coder provided judgments on 1470 comments. However, on average coders provided only 105 judgments. As a result, Fleiss Kappa and similar measures would be inappropriate for our data. However, as a means of exploring the robustness of our dataset, for each of profanity, insult, and insult object we fit a cross-classified mixed-effects model in which coders and items were random effects. Results showed small intra-class correlations for all three measures. This confirmed our intuition that the data was likely to be "noisy" because of the often-personal nature of profanity and insult-definitions. As Sheng and colleagues suggest (Sheng et al., 2008), employing multiple coders for each item can significantly improve quality, even with noisy data. As a result, each item was rated by a minimum of three raters. We adopted a simple consensus model to apply values for profanity, insult, and target of the insult, which we refer to as the "insult object." Four hundred ninety one comments (7.5%) were dropped from the final dataset because raters did not reach consensus on those items.

5. Experiments

An automated community management system relies on the ability to automatically identify negative content offered with malicious intent in a news group or forum. Towards such a system, we tested the performance of various configurations of systems aimed at determining whether a comment contains an insult, and whether or not that insult was directed at the author of a previous comment or at a third party (e.g. democrats, conservatives, Sarah Palin, Britney Spears, etc.). These experiments aim to surface not only whether such a semi-automated community management system is possible, but also which system configurations and features perform best, and how much labeled data is needed to reach optimal performance.

The systems employ techniques including support vector machines, known for their performance in text classification tasks (Joachims, 1998). Consistent with conclusions of others concerned with text classification tasks, we found that linear kernel support vector machines perform as well as polynomial, radial basis or sigmoid kernels (Zhang & Lee, 2003). As such, all support vector machines in our experiments employ linear kernels.

All systems are evaluated based on their precision-recall breakeven point, maximum f-measure (f1), and maximum accuracy averaged over 5 trials of 10-fold cross validation, as is standard practice for applications of this type (Joachims, 1998; Siersdorfer, Chelaru, Nejdl, & San Pedro, 2010; Yin et al., 2009). In particular, maximal accuracy alone would not be a strong measure of

classification performance with unbalanced class sizes, a characteristic of the problem at hand. That is, if 95% of the training data does not contain an insult, then a 95% accurate system can be achieved by simply classifying all comments negatively (as *not* containing an insult). The combination of maximal accuracy with maximal f-measure and precision-recall breakeven point conveys much more information about system performance and allows direct comparison to previous work. While these evaluations are very descriptive of the systems and their performance, it is important to note that in a *completely* automated online community management environment, we would likely value precision over recall to minimize type I errors (false positives) so that we do not mistakenly remove comments that are not actually offered with malicious intent. While we might not catch all inappropriate content, it would be more problematic to mistakenly remove legitimate comments, as that would discourage user participation. However, semi-automated approaches may value recall over precision to enable a more feasible human evaluation approach.

5.1 Detecting Insults

We trained a set of machine learning systems to detect the presence or absence of an insult in comments on a social news site. There are many variables that affect the performance of such a system. Which features and representation (presence or frequency) result in the most accurate classifier? Is it easier to detect insults within specific categories of comments (e.g., politics or entertainment) than with a general-purpose system? How does such a system perform against a baseline system and against the theoretical baseline (random classification and random classification weighted by the prior distribution of classes)? The following sections address these questions.

5.1.1 Features

Which features and representation (presence or frequency) result in the most accurate classifier of insults? To answer this question, we built a set of support vector machines trained on various combinations of and representations of term-based features. Features (using the bag-of-words approach) included unigrams, bigrams, stems and 'bi-stems' (stems of bigrams) and were represented with binary presence and with frequency of occurrence. The systems were trained on a corpus of 6009 comments (of the 6500 labeled comments, those which had a consensus of 60% or higher) – 1239 positive data points (comments that contain insults) and 4770 negative data points (comments that do not contain insults). That is, 20.62% of the dataset are comments that contain insults. The systems were trained and tested at varying 'cost-factors' (Morik, Brockhausen, & Joachims, 1999)– from 0.0 to 9.9 at 0.1 increments. In a support vector machine, the cost-factor indicates the factor "by which training errors on positive examples outweigh errors on negative examples" ("SVM-light documentation."). That is, altering the cost-factor of an SVM allows one to indicate whether minimizing type I errors (false positives) or type II errors (false negatives) is more important for a particular task. From these systems at various cost-factors, we calculated the precision-recall breakeven point, maximum accuracy and maximum f1 measure. The results are summarized in Table 2.

Table 2 includes 'random baseline' and 'weighted random baseline' systems. The 'random baseline' system simply labels target documents randomly, while the 'weighted random baseline' system labels target documents randomly, weighted by the distribution of positive and negative comments in the training set. Since these systems cannot be altered with a 'cost-factor', the chart

holds precision and recall values in lieu of a breakeven point. As expected, the precisions and f1 of both random systems approach the proportion of the dataset that is positive, while the recall and accuracy of the 'random baseline' system approach 0.5 as it will catch half of the positive cases. These values are merely shown to validate the theoretical baseline.

Though an accuracy of 0.8345 is encouraging (for the Stems – presence system), one must remember that only 20.62% of the dataset is positive, so an accuracy of 0.7938 can be attained by merely labeling all target comments as negative. However, we were very encouraged to achieve a precision-recall breakeven point of 0.5397 and maximal f1 of 0.5432 as both far surpass the baseline systems. Though the system using stems represented as binary presence features performed best, all combinations of features and representations performed similarly. That is, the choice of which term-based features (unigrams, bigrams, stems, bi-stems) and representation (frequency or presence) to use does not have a large impact on the system performance.

System - Features used (representation)	P-R BEP (or P/R)	Maximal F1	Maximal Accuracy
Stems (presence)	0.5397	0.5432	0.8345
Bigrams and stems (presence)	0.5317	0.5428	0.8333
Bigrams and stems (frequency)	0.5301	0.5414	0.8262
Stems (frequency)	0.5283	0.5373	0.8168
Random baseline	0.2030 / 0.4986	0.2885	0.4954
Weighted random baseline	0.2066 / 0.2090	0.2078	0.6730

Table 2: Evaluation of features and representations in insult classification.

For this particular task, there are other features available including features of the author of the comment (e.g. number of comments she/he has posted, length of time as a community member, number of times she/he has used a community voting mechanism) or other features of the comment itself (e.g. time the comment was posted, position in the thread). These features were intentionally avoided as the goal of this work is to build a system that can be easily ported to other sites. The choice of solely term-based features makes no assumptions about the availability of other features on another site. Optimizing system performance using only term-based features guarantees that this system can be used on other social news sites as it only assumes the presence of textual comments.

5.1.2 Categories

Is it easier to detect insults within specific categories of comments (e.g., politics or entertainment) than it is to do so in the context of the entire pool of comments? Intuitively, one might expect insult detection to be a domain-specific problem; the language that one might use to insult someone in a political domain might differ from that used in an entertainment or sports domain. If this is indeed a domain-specific problem, we could leverage domain information in order to build more accurate insult classifiers. Other factors, such as community demographics and normative

behaviors for specific sites, will undoubtedly also have an large impact on our ability to detect insults, however, the question at hand here is whether knowledge of the domain of each comment can improve the system's ability to detect insults.

Recall that the comments in our corpus are user contributions that follow news articles, images and videos on a news-story commenting site. In addition to the actual comments themselves, our corpus holds a large set of metadata including the time that the comment was posted, which 'parent' article it is in reference to, and a set of metadata surrounding the articles including its category. If the commenter replied directly to a previous comment, the 'parent' id for that article is the id of the previous comment (which in turn can of course be traced back to an article). Thus, for each comment, we can access the category (or list of categories) for the article that the comment is in reference to. This information can be used to slice our corpus of labeled comments into categorical sets. Note that because an article (and thus a comment) can be labeled with multiple categories, these categorical sets are not disjoint.

Within the metadata for an article, categorical information comes in the form of a list of categories for each article where a category is listed as *category_language-country* (e.g. politics_en-US, news_en-US). In our overall corpus of 1.6 million comments, parent articles span four languages (English, Spanish, French and Portuguese) and seven countries (France, US, Australia, Great Britain, Brazil, India, and Mexico). However, we limited the 6500 comments in our MTurk labeling task to US English. Within this labeled dataset, ten categories are represented: news, politics, entertainment, business, world, science, sports, health, lifestyle, and travel. While present in the dataset, the latter five did not occur enough to permit a viable evaluation (the last five categorical corpora contained 27, 21, 19, 10, and 2 data points respectively). Table 3 below shows a comparison of insult classification systems trained and tested within each category, compared to both their random baseline (which were, of course, consistent with the theoretical baseline and are merely included for completeness) and to a 'general' system trained on all data. All systems used stems represented by presence as features as those were the top performing from the previous section.

As shown in Table 3, the categorical insult classification systems all outperformed their random baseline. However, the general insult classification system (trained on data from all categories) outperformed all of the categorical insult classification systems. Looking at this evaluation, one might question whether performance was a function of the corpus size for each classifier. While not addressed directly for the task of insult detection, this question is addressed indirectly through an additional evaluation in section 5.2.3.

Category	**System**	**Corpus Size**	**Positive Data Points**	**Negative Data Points**	**Positive Ratio**	**P-R BEP (or P/R)**	**Maximal F1**	**Maximal Accuracy**
General (all data)	*Stems – presence*	6009	1239	4770	0.2062	0.5397	0.5432	0.8345
	Random baseline					0.2030/ 0.4986	0.2885	0.4954
News	*Stems – presence*	1728	371	1357	0.2147	0.4271	0.4713	0.7891

	Random baseline					0.2191/ 0.5108	0.3067	0.5022
Politics	*Stems – presence*	1569	420	1149	0.2677	0.5086	0.5226	0.7642
	Random baseline					0.2751/ 0.5158	0.3588	0.5062
Entertainment	*Stems – presence*	661	123	538	0.1861	0.4140	0.4191	0.8297
	Random baseline					0.1856/ 0.5032	0.2711	0.5162
Business	*Stems – presence*	451	75	376	0.1663	0.3455	0.3458	0.8384
	Random baseline					0.1821/ 0.5729	0.2764	0.5083
World	*Stems – presence*	443	84	359	0.1896	0.3175	0.3174	0.8034
	Random baseline					0.1855/ 0.4548	0.2635	0.5003

Table 3: An evaluation of the task of insult classification broken down by category.

5.2 Classifying the Insult Object

Next, we trained support vector machines to, given a comment that is known to contain an insult, detect whether that insult was directed at the author of a previous comment or at a third party. Again, there are many variables that affect the performance of such a system. Which features and representation (presence or frequency) result in the most accurate classifier? Is it easier to determine who the insult is directed toward when training and testing within specific categories of comments (e.g., politics or entertainment) than with a general-purpose system? How does the size of the training corpus affect the system accuracy? How does such a system perform against a baseline system and against the theoretical baseline (random classification and random classification weighted by the prior distribution of classes)? The following sections address these questions.

5.2.1 Features

Which features and representation (presence or frequency) result in the most accurate classifier? To answer this question, we performed the same type of evaluation described in section 5.1.1. Since we are now classifying insult object on comments that are known to contain insults, our dataset is smaller. The entire corpus of labeled insult object data (with a consensus of 0.6 or higher) contains 968 comments with 631 positive data points (comments that contain insults directed towards the author of a previous comment) and 337 negative data points (comments that contain insults directed toward a third party). The discrepancy between the 1239 positive data points in the insult corpus (that is 1239 comments with insults) versus the 968 total comments in this corpus can be explained by the 0.6 or higher consensus requirement; the insult object corpus only includes comments for which the insult *and* insult object labels from MTurk workers reached a consensus of 0.6 or higher, whereas the insult corpus only requires a consensus of 0.6 or higher for the insult label.

Comparing systems that utilize various features and representations, we found that bigrams and stems using a presence representation performed best, though again, all systems performed far better than the random baselines. Table 4 shows that the systems performed astonishingly well, reaching a peak precision-recall breakeven point and maximal f1 of 0.885. While these values are much higher than the same evaluation for insult classification, it should be noted that the random baselines for insult classification are far lower.

System - Features used (representation)	P-R BEP (or P/R)	Maximal F1	Maximal Accuracy
Stems (presence)	0.8810	0.8821	0.8450
Bigrams and stems (presence)	0.8850	0.8854	0.8484
Bigrams and stems (frequency)	0.8664	0.8681	0.8221
Stems (frequency)	0.8712	0.8720	0.8358
Random baseline	0.6589/0.5094	0.5746	0.5074
Weighted random baseline	0.6484/0.6464	0.6474	0.5402

Table 4: Evaluation of features and representations in insult object classification.

5.2.2 Categories

Is it easier to classify the insult object within specific categories of comments (e.g., politics or entertainment) than with a general-purpose system? To answer this question, we performed the same type of evaluation seen in section 5.1.2. As shown in Table 5, we do not see a significant improvement in system performance by limiting training and testing data to a single category. However, since the corpus was already somewhat small (968 data points), dividing it by category resulted in rather small training sets, which may have affected performance. This discrepancy is addressed in the next section. In this evaluation, the 'news' category did perform as well as the 'general' classifier and better in terms of maximal f1 whereas the news category performed worse in the corresponding insult detection evaluation in section 5.1.2.

Category	System	Corpus Size	Positive Data Points	Negative Data Points	Positive Ratio	P-R BEP (or P/R)	Maximal F1	Maximal Accuracy
General (all data)	*Bigrams and Stems – presence*	968	631	337	0.6519	0.8850	0.8854	0.8484
	Random baseline					0.6589/ 0.5094	0.5746	0.5074
News	*Bigrams and Stems – presence*	291	187	104	0.6426	0.8843	0.8915	0.8489
	Random baseline					0.6570/ 0.5052	0.5712	0.4926
Politics	*Bigrams and*	338	212	126	0.6272	0.8461	0.8499	0.8074

	Stems – presence							
	Random baseline					0.6416/ 0.4948	0.5587	0.4996
Entertainment	*Bigrams and Stems – presence*	93	57	36	0.6129	0.7365	0.8129	0.7478
	Random baseline					0.5506/ 0.4394	0.4888	0.4725
Business	*Bigrams and Stems – presence*	69	42	27	0.6087	0.5270	0.7360	0.6040
	Random baseline					0.6000/ 0.5000	0.5455	0.5200
World	*Bigrams and Stems – presence*	69	39	30	0.5652	0.7400	0.7727	0.7224
	Random baseline					0.4960/ 0.4697	0.4825	0.4571

Table 5: An evaluation of the task of insult object classification broken down by category.

5.2.3 Corpus Size

How does the size of the training corpus affect the system accuracy? In the previous section, we saw that a general-purpose insult object classification system outperformed those trained and tested within individual categories, however, the question lingered as to whether that was an effect of the size of the training corpus within the categories. To answer this question, we created several 'general' corpora of varying sizes to test whether or not the performance of a system trained on this data is altered by the size of the training set, and to what extent. By creating general corpora of varying sizes, we now have a fair comparison for each of the categorical classifiers. The results can be seen in Table 6.

Table 6 gives performance of insult classification systems (support vector machines using bigrams and stems as presence features) with rows in ascending order by corpus size. The 'General-n' classifiers are trained and tested on datasets generated by sampling n data points from the corpus of 6500 labeled comments. From these n comments, only those that met the 0.6 labeling consensus minimum requirement for both insult and insult object labels were used. From that set, the only useful data for this task are those comments that contain insults. For example, in the General-400 dataset, only 63 comments contain insults that met the consensus requirement, 38 of which are directed at the author of a previous comment ('positive') while 25 are directed at a third party ('negative'). The General-400 set was created as comparable to the business and world classifiers (corpus sizes of 63 and 69 data points respectively). The General-500 was intended to be comparable to the entertainment classifier, General-1500 and General-2000 to the news and politics classifiers. The boxes surrounding these rows are intended to convey these comparison sets. The results of the General-6500 (all data) system are merely included for completeness.

As expected, performance of the general purpose insult object classifiers do degrade as training corpus size decreases – excluding the discrepancy of precision-recall breakeven point for General-400 being slightly higher than General-500. While the business classifier performed worse than its comparable general classifier, the world classifier outperformed it. In fact, the world classifier outperformed both the General-400 and General-500 classifiers in all three performance measures (precision-recall breakeven point, maximal f1 and maximal accuracy). Similarly, entertainment and news surpassed their comparable general classifiers for all three measures, but politics performed only slightly better than its comparable general classifier. From this, one might conclude that 'world,' 'entertainment' and 'news' are categories that have domain-specific language used in their insults; language that contributes to the performance of insult object classifiers. For this reason, future work will include larger labeled corpora in these domains to test whether performance scales with corpus size.

Category	Corpus Size	Positive Data Points	Negative Data Points	Positive Ratio	P-R BEP (or P/R)	Maximal F1	Maximal Accuracy
General-400	63	38	25	0.6032	0.6660	0.7588	0.6160
Business	69	42	27	0.6087	0.5270	0.7360	0.6040
World	69	39	30	0.5652	**0.7400**	**0.7727**	**0.7224**
General-500	91	58	33	0.6374	0.6278	0.7598	0.6314
Entertainment	93	57	36	0.6129	**0.7365**	**0.8129**	**0.7478**
General-1500	226	148	78	0.6549	0.7903	0.8486	0.7955
News	291	187	104	0.6426	**0.8843**	**0.8915**	**0.8489**
General-2000	298	193	105	0.6477	0.8309	0.8535	0.7991
Politics	338	212	126	0.6272	0.8461	0.8499	0.8074
General-6500	968	631	337	0.6519	0.8850	0.8854	0.8484

Table 6: An evaluation of the task of insult object classification comparing classifiers of various training set sizes and categories.

5.3 Detecting Negative Content Offered with Malicious Intent

While the previous two sections (sections 5.1 and 5.2) describe systems that detect insults and classify the objects of the insults, neither of these systems on its own would suffice to surface inappropriate comments in online communities. In relation to our data, we make the assumption that an insult for which the object is the author of a previous comment is a negative comment offered with malicious intent. That is, detecting negative comments of malicious intent involves examining every comment in a forum or comment stream and finding those that contain insults for which the object is the author of a previous comment. You might consider this a two step classification (first, decide whether or not the comment contains an insult, and if it does, classify the object of that insult), however, one could also consider this a single binary classification problem where the positive class is comments that contain insults directed at the authors of previous comments, and the negative class is all other comments (including those comments that do not contain an insult, and those which contain an insult which is directed at a third party). We will compare methods that use both approaches.

In addition to the systems described in the previous two sections (5.1 and 5.2), we hypothesized that information about the valence of a comment, as well as knowledge about the comment's relevance to the original news article would prove useful in the detection of negative comments of malicious intent. To set the groundwork for our system, in the next sections (5.3.1 and 5.3.2), we outline in more detail our theoretical framework for the automated analysis through which we aim to surface relevance and valence.

5.3.1 Relevance Analysis

Recall that the comments in our corpus are user contributions that follow news articles, images and videos on a social news site. In addition to the actual comments themselves, our corpus holds the 'parent' that each comment is in reference to. If the 'parent' is an article, we have the full text of that article, which we'll call the focal story. In the case where we have a focal story (as opposed to an image or video), the relevance of the comment to the focal story could prove useful to our task of detecting negative comments of malicious intent. While the relevance of a comment to a video or image would also be useful, we utilize term-based relevance measures that require a textual article.

To determine whether or not each comment is *on-topic* (relevant to the focal story), we employ standard information retrieval techniques including *tfidf (Salton & Buckley, 1988)*. While it was originally created for early search engines as a method to evaluate the relevance of a document to a query, others have recently used this technique as a way to form a content query from a document to be passed off to a search engine to retrieve similar documents (Budzik, Hammond, & Birnbaum, 2001). In this way, it can be viewed as a document similarity metric.

The *tfidf* document representation technique treats each document as a bag of words, where words have varying degrees of importance in the document. The importance of a word is boosted by the frequency with which it occurs in the document (term frequency, TF). The importance of a word is inversely proportional to the frequency with which it occurs in documents across a corpus (document frequency, DF). A corpus of news stories from Reuters was used to compute document frequency values in our system ("Reuters-21578 text categorization test collection.").

Our system creates a *tfidf* representation of the focal story. It then treats each comment as a query and measures the relevance of the focal story to that query (comment). The relevance score is simply the sum of the *tfidf* values for each of the query terms in the focal story. If the relevance score is above a similarity threshold, the comment is considered to be *on-topic*. The real-valued outputs of such comparisons allow us to evaluate comments by degrees of similarity to the original focal story, in addition to simply *on-topic* or *off-topic.*

Table 7 shows several comments corresponding to the news story shown in Figure 1. Although there were many comments in the thread for this story, we report a sample here to illustrate our technique. Overall, several of the comments on the story could be interpreted as being flames or insults, which was one reason why we have selected it for illustrative purposes in this paper. In Table 7, we present a relevance score conveying how similar each comment was to the news story itself; that is, how on-topic the comment was. As noted above, the story, titled "Number One Reason Sarah Palin Is Smiling Today," is about Letterman's admission of infidelity, reporting

multiple affairs. Table 7 illustrates that the *tfidf* relevance scores clearly reveal which comments are on-topic; the top three comments in the table show relatively high *tfidf* scores while low scores are registered for posts that are clearly off-topic or are spam.

Comment	*TFIDF Relevance Score*
"The Palin family sucks, always have and always will. There are a few sex scandals in Sarah & Todd's past. Not to mention ethics problems etc. David is probably a nicer person than Sarah & Todd put together. She can smile at Dave's misfortune while conveniently forgetting her own sins."	5.239
"Say what you will. Palin rooked her naysayers. You can call her book whatever you can to dilute her success but, WHO'S LAUGHING NOW! The Clowns and Comics are presently in conference crying and consoling each other with their big towels wiping off their makeup and drying their big old shoes. Waaaaahhhhhh!"	3.754
"At least he apologized; Palin accepted his apology so I'll follow her lead. She was right about Letterman, which we all knew--and hopefully America realizes she has been right about OH so much more."	3.717
ClassyMingle.com is the best and largest online personals site dedicated to men and women seeking a higher caliber online dating experience.	0.359
Whyyyyyyyyyyyyyyy so many people are interested in an ageless relationship. young girls want to have fun with 40+ man and young guys want to have fun with 40+ women. There are many sites focusing on this kind of relationships such as http://www.Seekingsugar.com !	0.696
Who cares!!! My boyfriend thinks the same with me. He is eight years older than me, lol. We met online at Agelessmatch.com a nice and free place for Younger Women and Older Men, or Older Women and Younger Men, to interact with each other. Maybe you wanna check out or tell your friends.	0.351

Table 7: Comments and their tfidf relevance scores. High numbers are more relevant; low numbers less relevant. Very low relevance often indicates spam.

While these *tfidf* scores are very helpful in automatically detecting whether a post is or is not on-topic, we are also interested in detecting the emotional or affective trajectory of a comment. This is useful when used in combination with the *tfidf* measures of relevance because it allows us to determine whether a post is somewhat off-topic (median *tfidf* score), yet of a positive emotional type, which may indicate a conversational comment. As noted in related work, phatic statements and conversational comments are often the glue that moves a web site from being informational to being social (Gazan, 2009). Ideally in a community news comment site, we would want both of these elements.

5.3.2 Valence Analysis

As noted in section 2, we face several challenges in this task in the context of sentiment analysis, namely a lack of a sentiment labeled dataset of this type (user comments on news stories), the short and conversational nature of the comments, and moving beyond the valence dimension typically used in sentiment analysis. Thus far, we have moved beyond valence and labeled our dataset with the presence/absence of profanity, insults, and the object of the insult. However, we hypothesize that information about the valence of each comment might still be useful in determining if each comment is offered with malicious intent.

In section 2, we cite prior studies that outline why, and to what extent, sentiment analysis (valence analysis) is a domain-specific problem (among other dependencies). Thus, valence analysis systems need to be trained on data from within the target domain. When such training data is not available, some systems have been built in attempt to adapt to new domains without in-domain labeled training data (Anthony Aue & Michael Gamon, 2005). One such system, called *Reasoning Through Search*, was developed in past work by one of the authors and is a perfect choice for this situation, as our training data does not contain a valence label[3]. We use the *Reasoning Through Search* system as a method by which to gain valence judgments (a score ranging from -2 to +2) on each comment. This information will be used as an additional feature/signal in determining if a comment is of malicious intent.

The *Reasoning Through Search* system (*RTS*) is trained on a large set of movie and product reviews from widely varying domains (actors, books, colleges, destinations, drinks, electronics, food, movies, music, restaurants, software, sports and video games). The reviews are divided by domain on the site where they are posted (www.rateitall.com). To analyze a target document, the *RTS* system must first determine which of the 13 domains the document is most similar to. Next, the system characterizes the document with a "sentiment query" containing the words in the document with the highest "sentiment magnitude" (words that differ most in usage between 1 and 5 star reviews). This query is then posted to a case base of all labeled reviews (labeled with their star rating) to retrieve (emotionally) related documents. The labels of the retrieved documents are then combined with knowledge of the most similar domain of the target document in order to compute an overall valence intensity score – a score ranging from -2 to +2 where -2 is most negative and +2 is most positive (S. Sood et al., 2007).

RTS leverages domain relatedness in this task. That is, when attempting to analyze the valence of a document about a movie, if no training data exists within the movie domain, it will find the most similar domain and use its data (likely the books domain). While the *RTS* system does not utilize training data from the domain of comments following news stories, it is a good choice for this task given its ability to adapt to new domains. Thus, we use *RTS* to generate a valence label (a value between -2 and +2) for each comment. This information serves as an additional feature/signal in the multistep classification system described in the next section.

5.3.3 Multistep Classification

One of our approaches to building an end-to-end system to detect negative comments of malicious intent is to combine the classification systems presented in the previous sections. The first step involves passing the target comment through insult, and insult object classifiers (the best performing of the support vector machines described in the previous two sections (5.1 and 5.2)) as well as a valence analysis system (*RTS*) trained on movie and product reviews (described in section 5.3.2). Additionally, we retrieve the news article that the comment is referring to and perform relevance analysis using the tfidf approach (described in section 5.3.1) in order to obtain a relevance score – assessing how relevant the comment is to the original news article. These four systems serve as a first step in our multistep classification. The outputs of each system are then normalized to range from 0 to 1 and used as signals or features for another classifier which outputs a final decision (whether or not the comment is a negative comment of malicious intent). It is important to note that the classifier in the second step is trained via an additional

independent training set to avoid memorization of the training data. This multistep approach is compared against a single binary classification support vector machine using bigrams and stems as presence features - included as a baseline method.

The systems shown in Table 8 were all trained and tested on the complete labeled dataset of 6500 comments. Within this dataset, 6009 comments met the 0.6 labeling consensus requirement, 631 in the 'positive' class (containing insults that are directed at the author of another comment) and 5378 in the 'negative' class (comments not containing an insult at all, as well as comments containing an insult NOT directed at the author of another comment); the ratio of positive data points in the set was 0.1173. The experimental random baselines are included, though again, they approach the theoretical baselines as expected. Again, all results are averaged over 5 trials of 10 fold cross validation.

System - Features used (representation)	P-R BEP (or P/R)	Maximal F1	Maximal Accuracy
Multistep Classifier	0.5009	0.5038	0.9082
SVM (bigrams and stems – presence)	0.3781	0.3953	0.8980
Random baseline	0.1013/0.4953	0.1682	0.5005
Weighted random baseline	0.1007/0.1132	0.1066	0.8064

Table 8: An evaluation of the task of malicious content detection comparing classifiers of various approaches.

We see strong performance from the multistep classifier. This system outperforms the single support vector machine, supporting the need for a multistep approach, though both systems surpass the random baselines.

6. Conclusions

Through the experiments outlined in this article, we aimed to address and answer a set of research questions with regard to the feasibility of a tool to aid community managers in the detection of negative comments of malicious intent on social news sites. As an alternative to 'eyeballing' and/or relying on user 'abuse' reportage (social moderation), we have proposed, built and evaluated a detection system intended to aid community managers in automatically detecting negative comments of malicious intent. This tool is not intended to automatically remove inappropriate content, as different site have different social norms. Rather, this tool is intended to surface potential inappropriate comments that the community managers can then evaluate – greatly reducing the number of comments they must evaluate. In the section that follows, we address each of our research questions individually.

Can an automated system aid community managers in the task of identifying and removing malicious content? Surprisingly accurate systems emerged from our experiments – revealing that an automated system to aid community managers is more than promising. The best performing system is a multistep classifier that utilizes valence and relevance analysis systems, as well as a support vector machine trained with stems as binary presence features to detect insults and a

support vector machine trained with bigrams and stems as binary presence features to classify the object of the insults (these feature sets were chosen as they reached peak performance in experiments in sections 5.1.1 and 5.2.1). This multistep classifier far surpasses baseline performance at 0.5009 precision-recall breakeven point (baseline is 0.1173), 0.5038 maximal f1 (baseline is 0.1066), and maximal accuracy of 0.9082 (baseline is 0.8064). This system also fared better than a single support vector machine trained on the insult and insult object data to distinguish a comment containing an insult directed at the author of another comment, thus validating the need for inclusion of valence and relevance analysis systems. This system also outperforms previous work in this space. Recall from section 2 that our system (with a maximal f-measure of 0.5038) can be compared to the benchmarks created by Yin, et al. with maximal f-measures of 0.298 for detection of harassment on Slashdot, and 0.313 on MySpace (Yin et al., 2009).

What type, quality and quantity of training data are needed for such a system? There was a lack of availability of a training set of comments labeled by community managers; therefore we created one. 6500 comments from a popular news commenting site were labeled by Amazon Mechanical Turk workers for the presence/absence of profanity and insults, and the object of each insult. Normally, domain experts must do this sort of coding, but by using this MTurk labeled data to train our systems, we have shown that the quality of training labels generated by MTurk workers will suffice for this task, greatly reducing the overhead necessary to port this to other sites. We address training corpus size via an experiment in section 5.2.3. As expected, performance did degrade as the size of the training corpus decreased. However, gains in maximal f1 are minimal in increasing corpus size from 1500 to 2000 to 6500 comments (see Figure 5 for an illustration of this). For this reason, one might conclude that a minimal training corpus of 1500 labeled comments will suffice for this task.

Is this task domain/site specific - or will a single system suffice to reduce the scale of this problem for most sites? In sections 5.2.2 and 5.2.3, we showed that narrowing the detection problem down to the category level has the potential to increase system accuracy; that is, building separate systems to detect political comments of malicious intent (in comment streams following political stories), entertainment comment of malicious intent, etc. However, our labeled dataset was insufficient in size to do a full evaluation of this hypothesis. More labeled data within each category is needed. This raises the question of whether our system can generalize to other sites/domains. In this article, we limited the scope of our answer to this question to news categories within the single site we studied. In the future work section below, we expand on how this evaluation will continue to other sites.

7. Future Work

Future work includes evaluating the performance of our current system on test data from another site. Additionally, we'll also label a batch of data from the target domain using Amazon Mechanical Turk and compare performance to a system trained on data from the target site/domain. If the system performs better when trained on data from the target domain, the next step will be to find clusters of sites/domains based on detection system performance so that systems can be shared instead of requiring labeled training data from every new site. This is similar to an idea from the sentiment analysis community to leverage domain relatedness; that sentiment analysis systems

from 'similar' domains are interchangeable even though sentiment analysis is thought to be a domain-specific problem (Anthony Aue & Michael Gamon, 2005; Owsley, Sanjay Sood, et al., 2006; Pang & Lillian Lee, 2008). For example, a sentiment analysis system trained on movie reviews will perform reasonably well on book reviews, but poorly on restaurant reviews. We hypothesize that clusters of sites exist such that this strategy will yield sufficient detection systems for malicious content. By selecting domains the span much of Internet forum content, a set of systems may emerge that are able to handle domain-specific language from any site.

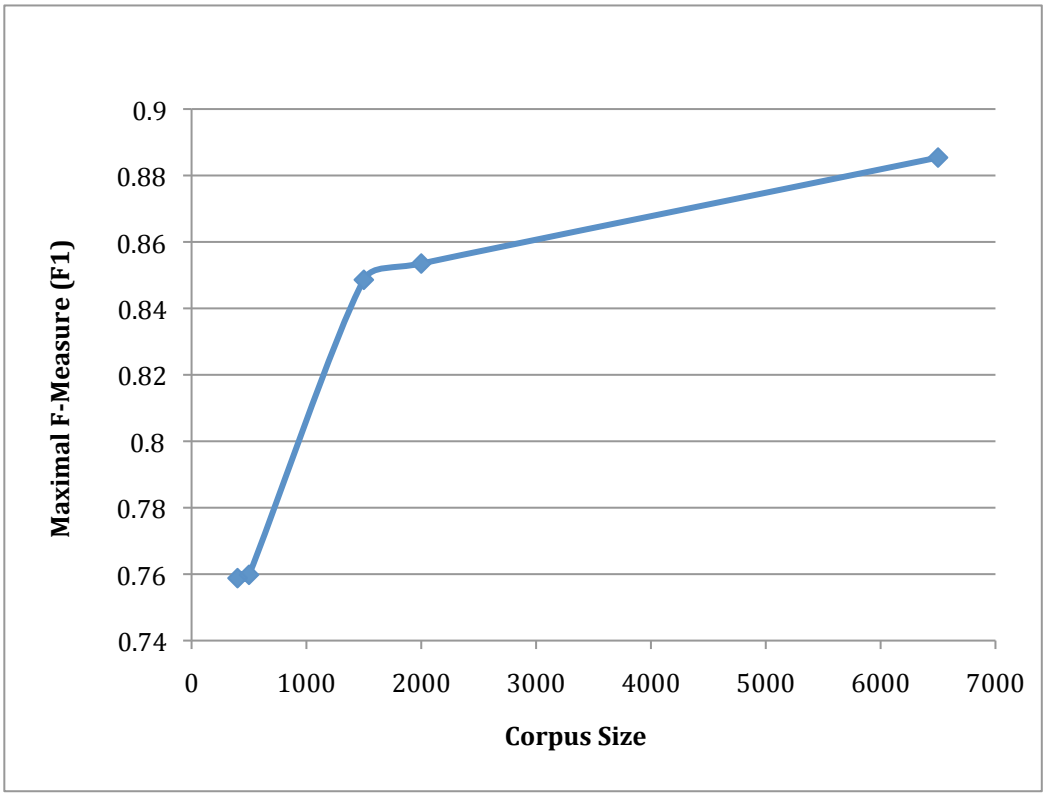

Figure 5: A graph of training corpus size against maximal f-measure achieved in the task of insult object classification.

Further future work involves investigating the impact of profanity in online forums. Can we detect profanity automatically and what effect does profanity have on participation? Does profanity incite profanity? Systems that detect profanity in text typically involve looking up words in a listing of profane terms. Once a term is found in that list, then one might proclaim that they have a positive case. However, recall for this baseline approach suffers in an online environment where users might disguise their profane terms (e.g., "@ss"), censor them (e.g., "#$%@"), use slang profanity (e.g., "assbite"), or misspell profane terms to give emphasis and/or avoid detection/censorship (e.g., "biatch", "shiiiiit"). These classes identify flaws for systems that rely on a corpus of profane terms. Can a profanity detection system trained on labeled comments outperform a baseline system that relies on a listing of profane terms by learning the context in which profanity occurs instead of the actual profane terms themselves? After training and evaluating new profanity detection systems, we plan to study how profane language effects conversations online.

Similarly, we plan to use our malicious content detection system in order to model trajectories

through comment threads. Specifically we wish to address the impact of these malicious comments on comments that follow: does negativity beget more negativity? (Sukumaran et al., 2011) Do conversations "go south"? And if they do, what are the characteristics of a conversation that escalates versus one that does not? Through this means, we intend to address the issue of whether undesirable behavior does or does not model undesirable behavior in others, and if so what are effective, in thread, remediation strategies which may be better than simple deletion. We believe these local strategies over content are the way to effective community management. With tools that help surface where non-socio-normative behaviors are occurring, we can support human community managers' work more effectively by automatically finding and filtering potential violations.

There are of course also open questions, especially when dealing with people who are regular community visitors. For example, an open question is about how a single person behaves over time: that is, what is the history of an author's sentiment about a topic over time? What is their authority in the social group, and from that what is their influence? Clearly some people's comments may have more weight than others'. Do we see the emergence of groups who all share the same sentiment on certain topics? These questions represent a valuable, but fine-grained and socially oriented research program.

8. Acknowledgments

We thank our colleagues at Yahoo! for their help with these analyses and the community managers for taking the time to talk with us, thus inspiring this work.

9. References

Anthony Aue, & Michael Gamon. (2005). Customizing Sentiment Classifiers to New Domains: a Case Study. *Proceedings of Recent Advances in Natural Language Processing - RANLP*. Presented at the Recent Advances in Natural Language Processing - RANLP. Retrieved from http://academic.research.microsoft.com/Publication/4112636/customizing-sentiment-classifiers-to-new-domains-a-case-study

Budzik, J., Hammond, K. J., & Birnbaum, L. (2001). Information access in context. *Knowledge-Based Systems*, *14*(1-2), 37-53. doi:doi: 10.1016/S0950-7051(00)00105-2

Callison-Burch, C. (2009). Fast, cheap, and creative: evaluating translation quality using Amazon's Mechanical Turk. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*.

Chesney, T., Coyne, I., Logan, B., & Madden, N. (2009). Griefing in virtual worlds: causes, casualties and coping strategies. *Information Systems Journal*, *19*(6), 525-548. doi:10.1111/j.1365-2575.2009.00330.x

Dinakar, K., Reichart, R., & Lieberman, H. (2011). Modeling the Detection of Textual Cyberbullying. *Proceedings of International AAAI Conference on Weblogs and Social Media, Workshop "Social Mobile Web."* Presented at the International AAAI Conference on Weblogs and Social Media.

Ekman, P. (2004). *Emotions Revealed: Recognizing Faces and Feelings to Improve Communication and Emotional Life* (First Edition.). Holt Paperbacks.

Faloutsos, M., Faloutsos, P., & Faloutsos, C. (1999). On power-law relationships of the Internet topology. *Proceedings of the conference on Applications, technologies, architectures, and*

protocols for computer communication, SIGCOMM '99 (pp. 251–262). Cambridge, Massachusetts, United States: ACM. doi:10.1145/316188.316229

Gamon, M., & Aue, A. (2005). Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms. *In Proceedings of the ACL 2005 workshop on feature engineering for machine learning in nlp.* (p. 57--64). Retrieved from http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.134.7821

Gazan, R. (2009). When Online Communities Become Self-Aware (pp. 1-10). Presented at the 42nd Hawaii International Conference on System Sciences, Big Island, HI. doi:10.1109/HICSS.2009.509

Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. In C. Nédellec & C. Rouveirol (Eds.), *Machine Learning: ECML-98* (Vol. 1398, pp. 137-142). Berlin/Heidelberg: Springer-Verlag. Retrieved from http://www.springerlink.com/content/drhq581108850171/

Kittur, A., Chi, E. H., & Suh, B. (2008). Crowdsourcing user studies with Mechanical Turk. *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08* (p. 453). Presented at the Proceeding of the twenty-sixth annual CHI conference, Florence, Italy. doi:10.1145/1357054.1357127

Laboreiro, G., Sarmento, L., Teixeira, J., & Oliveira, E. (2010). Tokenizing micro-blogging messages using a text classification approach. *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, AND '10 (pp. 81–88). Toronto, ON, Canada: ACM. doi:10.1145/1871840.1871853

Lampe, C., & Johnston, E. (2005). Follow the (slash) dot. *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work - GROUP '05* (p. 11). Presented at the the 2005 international ACM SIGGROUP conference, Sanibel Island, Florida, USA. doi:10.1145/1099203.1099206

Lampe, C., & Resnick, P. (2004). Slash(dot) and burn: distributed moderation in a large online conversation space. *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04 (pp. 543–550). New York, NY, USA: ACM. doi:10.1145/985692.985761

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, *10*(8), 707-710.

Lou, J. K., Chen, K. T., & Lei, C. L. (2009). A collusion-resistant automation scheme for social moderation systems. *IEEE Consumer Communications and Networking Conference, 2009.* (p. 571 -- 575).

M. M. Bradley, & P. J. Lang. (1999). *Affective norms for English words (ANEW): Stimuli, instruction manual and affective ratings.* (Technical No. C-1). Gainesville, FL: Center for Research in Psychophysiology, University of Florida. Retrieved from http://csea.phhp.ufl.edu/media.html

Mason, W., & Watts, D. J. (2009). Financial incentives and the "performance of crowds." *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '09 (pp. 77–85). Paris, France: ACM. doi:10.1145/1600150.1600175

Morik, K., Brockhausen, P., & Joachims, T. (1999). Combining Statistical Learning with a Knowledge-Based Approach - A Case Study in Intensive Care Monitoring. *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99 (pp. 268–277). Morgan Kaufmann Publishers Inc. Retrieved from http://portal.acm.org/citation.cfm?id=645528.657612

Owsley, S., Sood, S., & Hammond, K. J. (2006). Domain specific affective classification of documents. *Proceedings of the AAAI Spring Symposium on Computational Analysis of Weblogs* (p. 181 -- 183). Presented at the AAAI Spring Symposium on Computational Analysis of Weblogs.

Pang, B., & Lee, L. (2005). Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05 (pp. 115–124). Stroudsburg, PA, USA: Association for Computational Linguistics. doi:http://dx.doi.org/10.3115/1219840.1219855

Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, *2*(1-2), 1–135. doi:10.1561/1500000011

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02 (pp. 79–86). Stroudsburg, PA, USA: Association for Computational Linguistics. doi:http://dx.doi.org/10.3115/1118693.1118704

Reuters-21578 text categorization test collection. (n.d.).

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *INFORMATION PROCESSING AND MANAGEMENT*, *24*, 513--523.

Sheng, V. S., Provost, F., & Ipeirotis, P. G. (2008). Get another label? improving data quality and data mining using multiple, noisy labelers. *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '08* (p. 614). Presented at the Proceeding of the 14th ACM SIGKDD international conference, Las Vegas, Nevada, USA. doi:10.1145/1401890.1401965

Siersdorfer, S., Chelaru, S., Nejdl, W., & San Pedro, J. (2010). How useful are your comments?: analyzing and predicting youtube comments and comment ratings. *Proceedings of the 19th international conference on World wide web* (pp. 891–900).

Sood, S. O., & Churchill, E. F. (2010). Anger Management: Using Sentiment Analysis to Manage Online Communities. *Grace Hopper Celebration*.

Sood, S., & Vasserman, L. (2009). Esse: Exploring mood on the web. *Proceedings of the 3rd International AAAI Conference on Weblogs and Social Media (ICWSM)*.

Sood, Sanjay, Owsley, S., Hammond, K. J., & Birnbaum, L. (2007). *Reasoning Through Search: A Novel Approach to Sentiment Classification* (Technical No. NWU-EECS-07-05). Evanston, IL: Northwestern University. Retrieved from http://academic.research.microsoft.com/Publication/5467792/reasoning-through-search-a-novel-approach-to-sentiment-classification

Sukumaran, A., Vezich, S., McHugh, M., & Nass, C. (2011). Normative influences on thoughtful online participation. *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11 (pp. 3401–3410). New York, NY, USA: ACM. doi:10.1145/1978942.1979450

SVM-light documentation. (n.d.). Retrieved from http://svmlight.joachims.org/

Tetreault, J. R., Filatova, E., & Chodorow, M. (2010). Rethinking Grammatical Error Annotation and Evaluation with the Amazon Mechanical Turk. *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 45 - 48). Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.170.7443

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. *Proceedings of the 40th Annual Meeting on Association for*

Computational Linguistics, ACL '02 (pp. 417–424). Stroudsburg, PA, USA: Association for Computational Linguistics. doi:http://dx.doi.org/10.3115/1073083.1073153

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, *5*(2), 241-259. doi:16/S0893-6080(05)80023-1

Yin, D., Xue, Z., Hong, L., Davison, B., Kontostathis, A., & Edwards, L. (2009). Detection of Harassment on Web 2.0. *Proceedings of the Content Analysis in the WEB 2.0 (CAW2.0) Workshop at WWW2009*. Presented at the WWW, Madrid, Spain.

Zhang, D., & Lee, W. S. (2003). Question classification using support vector machines. *IN PROCEEDINGS OF THE 26TH ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMAION RETRIEVAL*, 26--32.

Footnotes:

1 http://www.nbcphiladelphia.com/news/politics/Number-One-Reason-Sarah-Palin-Is-Smiling-Today.html

2 We have replaced user IDs with 'C' for commenter, with the number denoting the specific commenter. We have used three dots '…' to denote where additional comments were made, that we felt were not useful for our discussion. These were usually off-topic exclamations or spam.

3 We attempted a pilot 'valence' coding of our data and found very low inter-rater reliability and thus, chose not to include 'valence' coding in the MTurk task.