Detecting Offensive Tweets via Topical Feature Discovery over a Large Scale Twitter Corpus

Guang Xiang Bin Fan Ling Wang Jason I. Hong Carolyn P. Rose
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA, 15213
{guangx, binfan, lingwang, jasonh, cprose}@cs.cmu.edu

ABSTRACT

In this paper, we propose a novel semi-supervised approach for detecting profanity-related offensive content in Twitter. Our approach exploits linguistic regularities in profane language via statistical topic modeling on a huge Twitter corpus, and detects offensive tweets using these automatically generated features. Our approach performs competitively with a variety of machine learning (ML) algorithms. For instance, our approach achieves a true positive rate (TP) of 75.1% over 4029 testing tweets using Logistic Regression, significantly outperforming the popular keyword matching baseline, which has a TP of 69.7%, while keeping the false positive rate (FP) at the same level as the baseline at about 3.77%. Our approach provides an alternative to large scale hand annotation efforts required by fully supervised learning approaches.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*; I.5.2 [**Pattern Recognition**]: Design Methodology—*Classifier design and evaluation*

General Terms

Algorithms, Languages, Human Factors

Keywords

Twitter, Hadoop, topic modeling, machine learning

1. INTRODUCTION

Social media sites represent some of the most popular sites on the Internet today. Offensive conent included in the user-generated content on many of these sites makes users' online experience unpleasant and may also be something that certain users want to filter (e.g. parents). Thus, an effective approach to detecting inappropriate online content is of great practical importance. In this paper, we opted to focus on profane language first, given that this is the most common category of inappropriate language in social media.

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

In our work we treat vulgarity as a type of linguistic style [11] that is expressed within a sentence with a certain rhythm or periodicity, which explains the tendency for a piece of vulgar text to contain more than one vulgar word. Another key component of our approach is leveraging a large text corpus with a high enough density of interesting patterns. The availability of a large enough corpus is more critical than the specifics of the algorithmic approach, especially in connection with data sets at the scale that is available over the web on social media sites [12]. In this work, we focus on Twitter, a popular microblogging service that provides a handy public platform for users to follow each other and share messages called tweets, which are text-based posts of up to 140 characters.

Our earlier results indicated that bag-of-words, part-of-speech (POS) and other pattern-based methods including belief propagation did not work well for profane tweet detection due to the significant noise in tweets. In this paper, we propose a hierarchical approach that exploits the co-occurrence of vulgar language via statistical topic modeling techniques and detects profane language with automatically generated features using a machine learning framework. In particular, we explore the predictive value of highly expressive topical features as well as reliable lexical features, and combine them in a single compact feature space.

In building the composite topical features, we first use a bootstrapping strategy to automatically collect a set of tweets from a number of offense-pro twitterers (i.e., twitterers who have a high prevalence of seed profane words) and law-abiding twitterers (i.e., twitterers who rarely use seed offensive words) over a large tweet corpus using a list of pre-defined offensive seed words; we then learn topic models from this tweet set via Latent Dirichlet Allocation (LDA) [3], a well-known generative topic modeling algorithm.

The major contribution of our proposed method is two-fold.

1. To the best of our knowledge, our work in this paper is the first computational approach to model profanity as a linguistic style in Twitter using a bootstrapping approach.

2. Our approach presented in this paper is the first to demonstrate that features induced using statistical topic modeling techniques for feature induction are more discriminative in a vulgarity detection task than a state-of-the-art keyword matching approach.

2. RELATED WORK

2.1 Prior Research on Offensive Language Detection

Offensive content detection is not a brand new area. Nevertheless, available techniques described in the literature are few and

mainly utilize a pre-defined dictionary of hostile words or patterns. In general, these approaches are rigid and crucially depend upon the coverage of the seed word or pattern list.

In the seminal work called Smokey [16], Spertus hand designed 47 features based on the syntax and semantics of the training sentences, catching 64% of the abusive messages in a testing set of 460 messages with an FP of 2%. Many of the features in Smokey suffer either from low coverage or a high false positive rate. Other work utilizing seed words or rules include [18, 10, 15].

2.2 Pattern-based Work on Social Media Analysis

In recent years, a variety of manual and automatic feature engineering techniques have been developed to construct feature spaces that are adept at capturing interesting language variation. Among them, two pattern-based approaches [6, 17] are similar to our own. However, given the extremely noisy and flexible nature of the Twitter messages, machine learning algorithms could be easily overwhelmed by a large set of trivial patterns.

2.3 Sentiment Analysis on Twitter

Twitter has drawn significant attention in recent years, and much work related to sentiment analysis has been published, such as [5, 4, 9]. Compared with offensiveness detection, the definition of a positive or negative attitude in sentiment analysis is relatively straightforward. For our task, the notion of vulgarity is rather subjective and the degree of offensiveness varies considerably among people, rendering the labeling process in our task even harder.

2.4 Bootstrapping Algorithms in NLP

Bootstrapping is a widely-used technique that provides labels given a large amount of unlabeled data and a small amount of seed information. Research that has applied bootstrapping to NLP problems has existed for more than a decade [8]. In this paper, we leverage bootstrapping to aid in vulgar tweet detection, and to the best of our knowledge, our work is the first in doing so. Furthermore, our target of offensive language is more subtle than many other tasks in the bootstrapping literature such as place name extraction [8].

3. ALGORITHMIC DETAILS

3.1 Twitter Corpus

The architecture of our system is given in Figure 1. Our tweet corpus contains the textual messages along with other meta data such as twitterer ID, posting time, etc. The raw tweets in our experiment were collected by querying the Twitter API as well as archiving the "Gardenhose" real-time stream [13]. Our raw corpus for training and testing purposes has more than 680 million and 16 million tweets respectively.

Tweets are expressed in an extremely colloquial fashion, with substantial noise and linguistic variation. For example, tweets contain a high volume of novel words, interjections, repetitions, orthographical errors such as word shortening (acronyms, words with characters removed, words shortened by phonetic spellings like *nite* for *night*), etc. Moveover, dropping spaces between words is also common, such as *howareyou*, which increases the scale of the tweet vocabulary significantly and imposes a huge burden for text analysis tasks.

3.2 Tweet Preprocessing

We designed a word cleaning algorithm, applying a series of filters in the following order to process the raw tweet corpus prior to topic induction and feature extraction.

1. We removed non-English tweets using LingPipe [1] with Hadoop.
2. To reduce the bias from heavy twitterers and increase diversity in learned patterns, we dropped tweets in the training set from twitterers with more than 1000 followers or followees.
3. We intentionally dropped retweets (indicated by "rt") from our training set, which refers to tweets from somebody else that one comes across and simply shares with others, because they unnecessarily magnify the weights of a certain words.
4. We removed the shortened URLs in tweets.
5. Twitterers often use mentions in the body of their tweets to refer to other people, which has the format of @username, and we removed these from the tweets.
6. The # symbol, called a hashtag, is used in Twitter to mark topics in a tweet, and we removed all hashtags from the tweets because a great volume of them are concatenated words, which tends to amplify the vocabulary size inadvertently and may hurt topic modeling.
7. To tackle the problem of intentional repetitions, we designed a heuristic to condense 3 or more than 3 repetitive letters into a single letter, e.g., *hhhheeeello* to *hello*. A similar heuristic has been used in other work such as [7].
8. For sequences of 2 repetitive letters, we counted how many such sequences each word in a tweet has, and condensed each such sequence into a single letter if the number of such sequences is over a threshold[1]. For example, *yyeeaahh* will be reduced to *yeah*, while *committee* remains intact.
9. We removed all stopwords.
10. We defined a word to be a sequence of letters, - or ', and removed all tokens not satisfying this requirement.

3.3 Feature Engineering

3.3.1 Topical Feature Construction

The idea is to treat each tweet as a finite mixture over an underlying set of topics, each of which is in turn characterized by a distribution over words, and then examine tweets via such topic distributions. Intuitively, offensive topics may be associated with higher probabilities for offensive words.

To learn a model that can infer topic distributions from tweets, we need a set of labeled training tweets with both offensive and non-offensive content, and to that end, we designed a bootstrapping algorithm to extract training tweets from a large tweet corpus using the map-reduce framework in Hadoop. The details are shown in algorithm 1, 2, and 3.

Our bootstrapping technique does not assume every word from the constant offenders to be offensive. Neither does it require a curse-free tweet set from benign twitterers. We expect topic modeling to pick up lexical collocation patterns in the profane content and produce meaningful topics for our task.

One merit of bootstrapping between twitterers and tweets is that with a limited list of seed words, we are able to capture a lot more novel offensive patterns automatically, thus tremendously reducing the effort in manual annotation.

With a set of training tweets as obtained in Algorithm 1, we adopt Latent Dirichlet Allocation (LDA) [3], a renowned generative probabilistic model for topic discovery, to build the composite topical features. We chose the LDA implementation by Phan et al.[14].

[1]We used a threshold of 3 in this work.

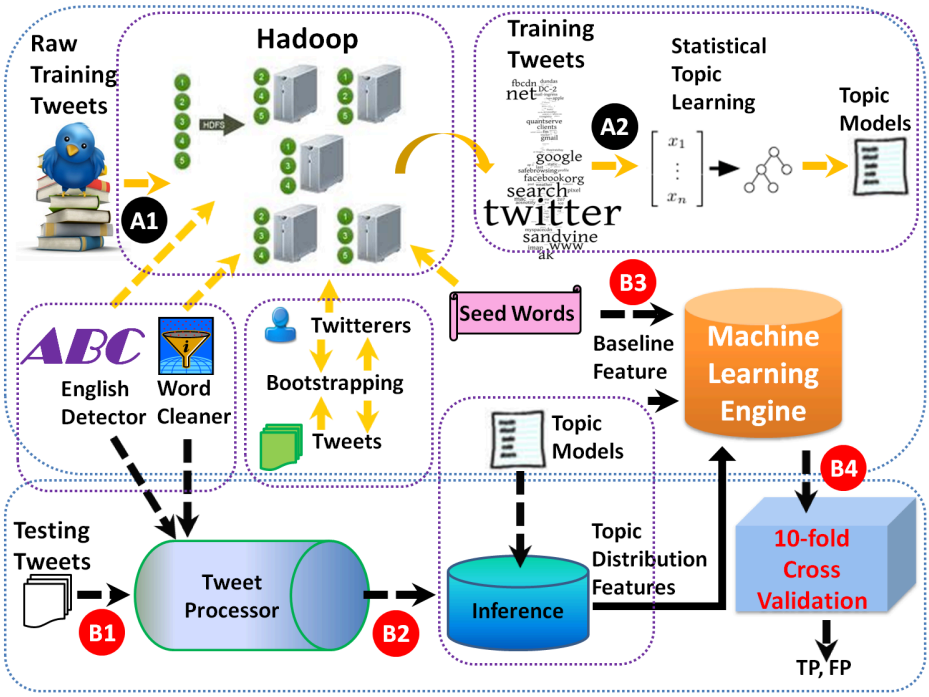

Figure 1: System architecture. A1) Bootstrap between twitterers and tweets based on a seed word set to obtain training tweets for topic model learning; A2) topic models are learned via a generative LDA approach; B1) tweets in a holdout testing set are processed in the same fashion as in A1); B2) topic distributions are inferred for each testing tweet by the topic model learned in step A2; B3) seed words are applied against each testing tweet, leading to a binary lexicon feature; B4) ML models are built and evaluated.

Algorithm 1 BuildTrainingTweetsViaBootstrapping

Require: raw tweets T, threshold t, seed words S
Ensure: tweet set TS for topic learning
1: $ot, gt \leftarrow$ ClassifyTwitterers(T, t, S)
2: $TS \leftarrow$ ExtractTweets(T, ot, gt)
3: **return** TS

Algorithm 2 ClassifyTwitterers

Require: raw tweets T, threshold t, seed words S
Ensure: offensive twitterers ot, good twitterers gt
1: preprocess T with the English detector and word cleaner by Hadoop
2: compute the percent p of offensive tweets for each twitterer based on S by Hadoop
3: $ot \leftarrow$ twitterers with $p >= t$
4: $gt \leftarrow$ twitterers with $p = 0$
5: **return** ot, gt

Algorithm 3 ExtractTweets

Require: raw tweets T, offensive twitterers ot, good twitterers gt
Ensure: tweet set TS for topic learning
1: preprocess T with the English detector and word cleaner by Hadoop
2: $OT \leftarrow$ get all tweets from each twitterer in ot by Hadoop
3: $GT \leftarrow$ randomly sample $|OT|$ tweets from gt by hadoop
4: $TS \leftarrow OT + GT$
5: **return** TS

3.3.2 *The Lexicon Feature*

The keyword matching technique, though narrow in coverage, can catch common vulgar language (with false positives sometimes depending on the context) and we exploit this property to introduce a lexicon feature into our framework, which is a binary indicator that there is at least one word from our offensive lexicon in the tweet. With the two types of features, our approach builds machine learning models to classify tweets, as shown in Algorithm 4.

Algorithm 4 ClassifyTweets

Require: tweet set TS for topic learning, testing tweet set TT, labels L for TT, seed words S
Ensure: classification result set r
1: $m \leftarrow$ learn topic models with LDA on TS
2: $F \leftarrow \phi$
3: **for** each tweet t in TT **do**
4: $f_t \leftarrow$ infer the topic distributions with m
5: $f_b \leftarrow$ check whether t has an offensive word in S
6: $f \leftarrow$ build a feature vector concatenating f_t, f_b
7: $F \leftarrow F \cup f$
8: **end for**
9: $r \leftarrow$ do 10-fold cross validation on F with respect to L
10: **return** r

4. EXPERIMENT

4.1 Experimental Settings

We compiled a dictionary of 338 most common offensive words based on [2], and manually removed entries used often yet not very offensive such as "hell". Moreover, we used the tweets crawled

from May 25, 2009 to October 17, 2010 as the raw training corpus for topic model learning, and took tweets from October 18, 2010 to October 27, 2010 as the raw testing corpus from which we selected testing tweets for the final evaluation. Table 1 gives some basic statistics about these two raw corpus.

For evaluation, we randomly sampled a subset of 4029 tweets from the raw testing corpus. To guarantee enough offensive patterns in this testing set, we first computed the percentage of offensive tweets p from each twitterer in the raw testing corpus based on our seed lexicon, and then randomly chose tweets from twitterers with $p >= 40\%$. We then recruited three participants on campus with different backgrounds to label the tweets in this testing set.

Table 1: Statistics of our raw training and testing corpus. 4.58% tweets in the raw training tweet corpus contain at least one offensive word in our seed list.

	Training corpus	Testing corpus
#Tweets	680,803,805	16,385,084
#Retweets	67,047,221	2,237,468
#URLs	118,941,407	2,728,363
#@usernames	260,599,517	6,198,205
#Hashtags	81,561,580	2,354,449
#Stopwords	4,645,096,146	108,785,278
#Other words	4,244,592,171	101,020,079

4.2 Experimental Results

To fully evaluate our approach, we adopted 4 popular machine learning algorithms, including J48 decision tree learning, Support Vector Machines (SVM), logistic regression (LR) and random forest (RF). We found that these 4 algorithms performed competitively, and due to the limitation of space, we only report the result of LR, which slightly outperformed the others. Moreover, though the volume of the raw training corpus is huge (Table 1), the tweet set produced by our bootstrapping technique (Algorithm 1), which was actually used to learn the topic model, has only 860,071 tweets, a tiny number compared with the raw training corpus. Therefore, we chose to learn a small number of topics (from 10 to 50) via LDA.

Table 2 shows the F1 values of LR under 10-fold CV using a threshold of 0.5 on the predicted probabilities. The F1 values improved as the number of learned topics increased, which is reasonable because we can catch more fine-grained patterns and thus better separate offensive and non-offensive content with more topics. Our approach using the full feature set outperformed the keyword matching baseline under all configurations with all machine learning algorithms, suggesting the robustness of our approach across various learning schemes.

In addition, TP and FP are two important metrics in evaluating binary classification tasks. We chose a threshold on the predicted probabilities in LR such that the resultant FP of our approach is on the same level as the keyword matching baseline, and plot the TP of our approach using LR in Figure 2. The graph shows that under such configurations, our approach using the full feature space improves the TP over the baseline by up to 5.4%. The ROC curves in Figure 2 also indicate that our approach with the full feature space has superior performance, dominating the algorithm with topic features only. Even the keyword matching baseline works reasonably well, and existing techniques rely to a great extent on it in detecting vulgarity. The experiment results here suggest that our approach is able to detect up to 5.4% more profane patterns without sacrificing the FP, which is a statistically significant improvement and is of great practical importance. Moreover, we can always tune the threshold on the predicted probabilities and other parameters to achieve a desirable detection rate, depending on the specific needs to go more aggressively or conservatively against offensive language.

4.3 Error Analysis

To further understand our technique, we conducted an error analysis after the experiment using 10 topics.

The keyword matching baseline utilizes a lexicon of offensive words compiled by us, however, 101 testing tweets with at least one word appearing in our seed lexicon were assigned a label of "not-offensive" by the human labelers we recruited. This led to all the 3.77% false positives by the baseline algorithm, which indirectly propagated to the lexicon feature of our proposed approach. This again confirms that the degree of offensiveness is rather subjective, and the task of offensiveness detection is difficult.

Moreover, we found that two features, i.e., the lexicon feature and topic 6, are responsible for many false positives. This comes as no surprise in that topic 6 contains the most offensive terms, either known ones in the seed lexicon or novel ones. On the other hand, no topics had a significant impact on the false negatives, and we conjecture that false negatives were caused by an additive effect of the topical features except for topic $4, 6, 9$.

5. DISCUSSION

Our approach can be further improved as follows. First, we utilized the topical features only via the word level distributions to detect frequently co-occurring patterns, while in the future we could consider the benefits of more complex features within the topic representation. This is consistent with our finding that a great many unigrams in topic 4 learned by LDA over the training tweets are not offensive by themselves, but are clearly indicative of sex-related offense when combined with other words, such as "wet", "dirty", etc. Second, we currently simply used a binary lexicon feature to capture the appearance of profane words for each tweet, and alternatively, we could adopt a complex weighting mechanism like TF-IDF. Third, the current tweet set for training the topic model has 860,071 tweets only. With more data, we can learn more topics via model tuning.

6. CONCLUSIONS

In this paper, we propose an approach that exploits the lexical collocation of profane language via statistical topic modeling techniques and detects offensive tweets using highly expressive topical features as well as the reliable lexicon feature in a single machine learning framework. The keyword matching technique has been shown to perform very well in the literature and achieved a TP of 69.7% with an FP of 3.77% in our experiment. While keeping the FP on the same level as the baseline, our approach had a TP of 75.1% over 4029 testing tweets using Logistic Regression, a significant 5.4% improvement over the baseline. In addition, our approach also provides an alternative to large scale hand annotation efforts required by supervised learning approaches.

7. REFERENCES

[1] Alias-i. Lingpipe 4.0.1. 2008. http://alias-i.com/lingpipe.

[2] AllSlang. List of swear words. 2010. http://www.noswearing.com/dictionary.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Table 2: The F1 values of Logistic Regression using a threshold of 0.5 on the predicted probabilities. Augmented by the lexicon feature, our approach outperformed the keyword matching baseline under all configurations.

		#topics learned by LDA on the training data				
ML algorithm	lexicon feature	10	20	30	40	50
Logistic Regression	NO	0.648	0.712	0.745	0.739	0.746
	YES	0.825	0.834	0.835	0.841	0.849
Keyword matching baseline		0.787				

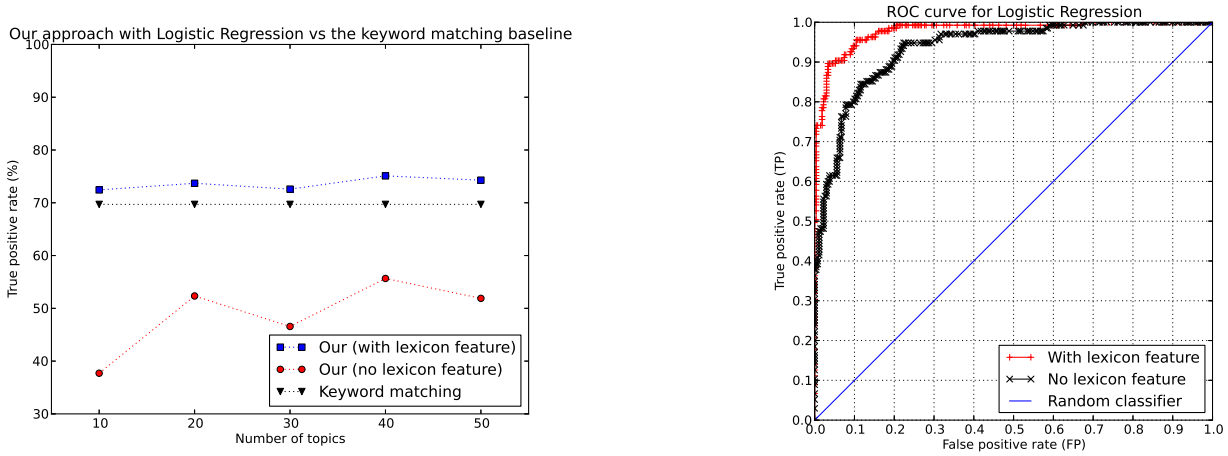

Figure 2: Using a threshold of 0.71 on the predicted probabilities in making the binary classification, Logistic Regression under the full feature representation improves the TP (left) over the keyword matching baseline by up to 5.4% (50 topic features) while keeping the FP on the same level. The ROC curve (right) of Logistic Regression using 50 topic features further shows the effectiveness of our technique in classifying swearing tweets.

[4] J. Bollen, H. Mao, and A. Pepe. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM'11)*, 2011.

[5] D. Davidov, O. Tsur, and A. Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*, 2010.

[6] P. Gianfortoni, D. Adamson, and C. Rose. Modeling of stylistic variation in social media with stretchy patterns. In *Proceedings of the First Workshop on Algorithms and Resources for Modelling of Dialects and Language Varieties*, pages 49–59. Association for Computational Linguistics, 2011.

[7] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. Technical report, Stanford University, 2009.

[8] R. Jones, A. Mccallum, K. Nigam, and E. Riloff. Bootstrapping for text learning tasks. In *Proceedings of the Workshop on Text Mining: Foundations, Techniques and Applications in the Sixteenth International Joint Conference on Artificial Intelligence*, pages 52–63, 1999.

[9] E. Kouloumpis, T. Wilson, and J. Moore. Twitter sentiment analysis: The good the bad and the omg! In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM'11)*, 2011.

[10] A. Mahmud, K. Z. Ahmed, and M. Khan. Detecting flames and insults in text. In *Proceedings of the Sixth International Conference on Natural Language Processing*, 2008.

[11] J. R. Martin and P. R. White. *The Language of Evaluation: Appraisal in English*. Palgrave Macmillan, 2005.

[12] P. Norvig. Statistical learning as the ultimate agile development tool. In *ACM 17th Conference on Information and Knowledge Management (CIKM'08)*, 2008.

[13] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, 2010.

[14] X.-H. Phan and C.-T. Nguyen, 2007. GibbsLDA++: A C/C++ implementation of latent Dirichlet allocation (LDA).

[15] A. H. Razavi, D. Inkpen, S. Uritsky, and S. Matwin. Offensive language detection using multi-level classification. In *Proceedings of the 23rd Canadian Conference on Artificial Intelligence*, pages 16–27, 2010.

[16] E. Spertus. Smokey: Automatic recognition of hostile messages. In *Proceedings of the Ninth Conference on Innovative Applications of Artificial Intelligence*, pages 1058–1065, 1997.

[17] O. Tsur, D. Davidov, and A. Rappoport. Icwsm - a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media (ICWSM'10)*, pages 162–169, 2010.

[18] Z. Xu and S. Zhu. Filtering offensive language in online communities using grammatical relations. In *Proceedings of the Seventh Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference*, 2010.