Detecting Nastiness in Social Media

Niloofar Safi Samghabadi♠, Suraj Maharjan♠, Alan Sprague♣,
Raquel Diaz-Sprague♣, Thamar Solorio♠
♠Department of Computer Science, University of Houston
♣Department of Computer & Information Sciences, University of Alabama at Birmingham
nsafisamghabadi@uh.edu, smaharjan2@uh.edu,
sprague@cis.uab.edu, diazspra@uab.edu, tsolorio@uh.edu

Abstract

Although social media has made it easy for people to connect on a virtually unlimited basis, it has also opened doors to people who misuse it to undermine, harass, humiliate, threaten and bully others. There is a lack of adequate resources to detect and hinder its occurrence. In this paper, we present our initial NLP approach to detect invective posts as a first step to eventually detect and deter cyberbullying. We crawl data containing profanities and then determine whether or not it contains invective. Annotations on this data are improved iteratively by in-lab annotations and crowdsourcing. We pursue different NLP approaches containing various typical and some newer techniques to distinguish the use of swear words in a neutral way from those instances in which they are used in an insulting way. We also show that this model not only works for our data set, but also can be successfully applied to different data sets.

1 Introduction

As the internet has become the preferred means of communication worldwide[1], it has introduced new benefits as well as new dangers. One of the most unfortunate effects of online interconnectedness is Cyberbullying – defined as the deliberate use of information/communication technologies (ICT's) to cause harm to people by causing a loss of both self-esteem and the esteem of their friendship circles (Patchin and Hinduja, 2010). The groups most affected by this phenomenon are teens and pre-teens (Livingstone et al., 2010). According to a High School Youth Risk Behavior Survey, 14.8% of students surveyed nationwide in the United States (US) reported being bullied electronically (nobullying.com, 2015). Another research done by the Cyberbullying Research Center (Patchin, 2015) from 2007 to 2015 shows that on average, 26.3% of middle and high school students from across the United States have been victims of cyberbullying at some point in their lives. Also, on average, about 16% of the students have admitted that they have cyberbullied others at some point in their lives. Studies have shown that cyberbullying victims face social, emotional, physiological and psychological disorders that lead them to harm themselves (Xu et al. (2012)).

In this research we perform the initial step towards detecting invective in online posts from social media sites used by teens, as we believe it can be the starting point of cyberbullying events. We first create a data set that includes highly negative posts from ask.fm. ask.fm is a semi-anonymous social network, where anyone can post a question to any other user, and may choose to do so anonymously. Given that people tend to engage in cyberbullying behavior under the cover of anonymity (Sticca and Perren, 2013), the anonymity option in ask.fm, as in other social media platforms, allows attackers the power to freely harass users by flooding their pages with profanity-laden questions and comments. Seeing a lot of vile messages in one's profile page often disturbs the user. Several teen suicides have been attributed to cyberbullying in ask.fm (Healy, 2014; Shute, 2013). This phenomenon motivated us to crawl a number of ask.fm accounts and to analyze them manually to ascertain how cyberbullying is carried out in this particular site. We learned that victims have their profile page flooded with abusive posts. Then from identifying victims of cyberbullying, we switched to looking for word patterns

[1]The New Era of Communication Among Americans http://www.gallup.com/poll/179288/new-era-communication-americans.aspx

Proceedings of the First Workshop on Abusive Language Online, pages 63–72,
Vancouver, Canada, July 30 - August 4, 2017.

that make a post abusive. Since, abusive posts are rare compared to the rest of online posts, in order to ensure that we would obtain enough invective posts, we decided to focus exclusively on posts that contain profanity. This is analogous to the method used in data collection by Xu et al. (2012); they limited their Twitter data to tweets containing the words *bully, bullied, bullying*.

The main contributions of this paper are as follows: We create a new resource to investigate negative posts in a social media platform used predominantly by teens, and make our data set public. The most noticeable difference of our data set with previous similar corpora is that it provides a generalized view of invective posts, which is not biased towards a specific topic or target group. In our data, each post is judged by three different annotators. Then we perform experiments with both typical features (e.g. linguistic, sentiment and domain related) and newer features (e.g. embedding and topic modeling), and combinations of these features to automatically identify potential invective posts. We also show the robustness of our model by evaluating it on different data sets (Wikipedia Abusive Language Data Set, and Kaggle). Finally, we do an analysis of bad word distributions in our data that, among other things, reflects a sexualized teen culture.

2 Related Research

Since our research goal is to detect nastiness in social media as an initial step to detect cyberbullying, we analyze previous works focusing on cyberbullying detection. Researchers (Macbeth et al., 2013; Lieberman et al., 2011) have reported that cyberbullying posts are contextual, personalized and creative, which make them harder to detect than detecting spam. Even without using bad words, the post can be hostile to the receiver. On the other hand, the use of negative words does not necessarily have a cyberbullying effect (al-Khateeb and Epiphaniou, 2016). Researchers have used different approaches to find cyberbullying traces.

Dinakar et al. (2012) concentrate on sexuality, race and culture, and intelligence as the primary categories related to cyberbullying. Then, they construct a common sense knowledge base - BullySpace - with knowledge about bullying situations and a wide range of everyday life topics. The overall success of this experiment is 66.7% accuracy for detecting cyberbullying in YouTube comments. Xu et al. (2012) identify several key problems in using the social media data sources to study bullying and formulate them as NLP tasks. In one of their approaches, they use latent topic modeling to analyze the topics people commonly talk about in bullying comments, however they find most topics were hard to interpret. Van Hee et al. (2015) study ask.fm Dutch posts, and develop a new scheme for cyberbullying annotation based on the presence and severity of cyberbullying, the role of the post's author, and a number of fine-grained categories associated with cyberbullying. They use the same two class classification tasks as the previous studies to automatically detect cyberbullying posts and achieve an F-score of 55.39%. Kansara and Shekokar (2015) combine text and image analysis techniques and propose a framework for detecting potential cyberbullying threats that analyze texts and images using a bag of words and a bag of visual word models respectively.

There is also some research in the field of online harassment and hate speech detection. Yin et al. (2009) apply a supervised machine learning approach to the automatic detection of online harassment. They combine content, sentiment, and contextual features and achieve an F-score of 44%. Nobata et al. (2016) use data gathered from Yahoo! Finance and News, then present a hate speech detection framework using n-gram, linguistic, syntactic and distributional semantic features and get an F-score of 81% for a combination of all features.

In this study, we present a data set containing question-answer pairs from ask.fm, which are labeled as positive (neutral), or negative (invective). Our data is conversational data from teenagers. We also have metadata containing information about the users that eventually can help us to focus on users who are being bullied with frequent profanity and also in analyzing the patterns used by attackers. Moreover, compared to previous work, we apply a wider range of different types of typical and newer NLP features, and their combinations to improve the classification performance. Following this approach, we reach F-scores of 59% for identifying invective posts in our own data set. Applying our classification model on Kaggle and Wikipedia data (we will introduce them later) shows that our method is robust and applicable to

other data. We also do an analysis of bad word distribution in our data set that shows that most of these bad words are often used in a casual way, so detecting cases in which there are potential invective requires careful feature engineering.

3 Data Collection and Annotation

Since most of the abusive posts we observed on our small scale study contained profanities, we decided to analyze the occurrence of bad words in a random collection of social media data. We crawled about 586K question-answer pairs from 1,954 random users in ask.fm from 28th January - 14th February, 2015. We limited crawling to posts in English by determining the percentage of English words ($\geq 70\%$) in the user's first page with an English dictionary (pyenchant [2]).

To create our bad words list, we compiled a list from Google's bad words list [3] and words listed in (Hosseinmardi et al., 2014). Based on frequency of use of each bad word in the list, we shortlisted some of them and their morphological variations and slang. We then looked at a small sample of data and filtered all posts containing any of these bad words. The resulting data set contains about 350 question-answer pairs. This small portion of data was divided among 5 different annotators for two-way annotation and disagreements were resolved by a third annotator. From these annotations, we computed the negative use rate (NUR) of each bad word (w_i). Equation 1 defines NUR. $Count(PI, w_i)$ and $Count(PN, w_i)$ are the counts of posts containing word w_i tagged as *invective* and *neutral* respectively.

$$NUR(w_i) = \frac{Count(PI, w_i)}{Count(PI, w_i) + Count(PN, w_i)} \quad (1)$$

According to NUR, we ranked the list of foul words, and removed words which were below the threshold (0.05). The final list includes the words *f*ck*, *a*s*, *sh*t*, *die*, *kill*, *h*e*, *as**ole*, *s*ck*, *n**ger*, *stfu*, *b*tch*, and *cut*. We called this small set of annotated data as "gold data" and use it for annotating a larger sample of data via CrowdFlower [4].

[2] http://pythonhosted.org/pyenchant/

[3] https://code.google.com/p/badwordslist/downloads/detail?name=badwords.txt

[4] http://www.crowdflower.com/

3.1 Crowdsourcing Annotations

With our small gold annotated data, we started a crowdsourcing task of annotating around 600 question-answer pairs in CrowdFlower. We provided a simple annotation guideline for contributors with some positive and negative examples to ease their task. Each question-answer pair was annotated by three different contributors. Figure 1 shows the interface we designed for the task.

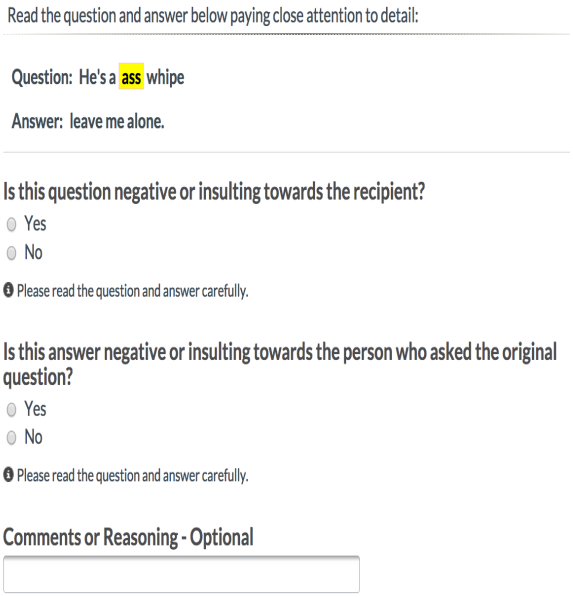

Figure 1: CrowdFlower interface to contributors

For ensuring high quality of the data, the same data was reviewed and annotated by 4 in-lab annotators using a two way annotation scheme. Initially, we found that the inter-annotator agreement was low. Hence, we changed the annotation guideline until the contributors and our internal annotators had reasonable agreement. We learned that although the task may seem simple, it may not so for the external contributors. Thus, it is necessary to iterate the process several times to ensure high quality data. Then, from the original set containing our gold data and extra 600 labeled pairs, we labeled more data with a combination of in-lab and crowdflower annotations into two classes: invective and neutral. Eventually, with this iterative process we annotated around 5,600 question and answer pairs. The average in-lab inter annotator agreement kappa score is 0.453. Table 1 shows the final data distribution. The data can be accessed via our website [5].

[5] http://ritual.uh.edu/resources/

Class	Question	Answer	Total
invective	1,114	909	2,023
neutral	4,483	4,688	9,171
Total	5,597	5,597	11,194

Table 1: Statistics for our ask.fm data

3.2 What is in the Data?

While annotating, we found instances of sexual harassment directed towards female users. Example 1 in Table 2 shows this type of abuse. In most of these cases, the attacking user is anonymous and he/she is constantly posting similar questions on the victim's profile.

We also found several instances where the purpose of the post is to defend/protect self or another person by standing up for a friend or posting hostile or threatening messages to the anonymous users (Example 2 in Table 2). This kind of post indirectly suggests that the user is being cyberbullied.

Also, the use of profane words does not necessarily convey hostility. In Example 3 in Table 2, looking at the question and answer pair, it is obvious that they are joking with each other.

In ask.fm, there are users that discourage cyberbullying by listening to the victims' feelings and motivating them to stay strong and not to hurt or kill themselves. Example 4 in Table 2 illustrates this case.

Ex.	Posts
1	***Question*** *Send nudes to me babe? :) I'll send you some :)* ***Answer****: stfu* ***Question****: C'mon post something sexy. Like a yoga pants pic or your bra or thong*
2	***Question****: She's not ugly you blind ass bat*
3	***Question****: you + me + my bed = fuckkk (;* ***Answer****: Haha oooooh shit (;*
4	***Question****: well I just want you to know I'm suicidal and 13. and I'm probably gonna kill myself tonight . . .* ***Answer****: No please don't seriously god put you on this earth for a reason and that reason was not for you to take yourself off of it . . .*

Table 2: Examples of different topics in our data set

The above examples show that our data set covers a wide range of topics related to cyberbullying. We believe that the data set will be a resource for other researchers carrying out abusive language detection research.

3.3 Comparison with the Other Data sets

Kaggle data released in 2012 for a task hosted by Kaggle called Detecting Insults in Social Commentary [6]. This data contains posts on adult topics like politics, employment, military, etc. Compared with ours, the Kaggle data is more balanced (26.42% of data labeled as insult). Wikipedia abusive language data set (Wulczyn et al., 2016) includes approximately 115k labeled discussion comments from English Wikipedia. The data set was labeled via Crowdflower annotators on whether each comment contains a personal attack. Only 11.7% of the comments in this data set were labeled as personal attacks. Table 3 compares the average length of the posts and words between our data and two other data sets. As we can see in this table, posts in ask.fm are much shorter than Kaggle and Wikipedia data. It also seems that users in ask.fm tend to use shorter words or even more abbreviations.

Avg length of post	ask.fm	Kaggle	Wikipedia
Avg no. of words	13.92	38.35	81.29
Avg length of words	4.73	5.54	5.94

Table 3: Average length of the posts, and words in ask.fm, Kaggle, and Wikipedia data sets in terms of the average number of words and the average number of characters

4 Methodology

In this work, we apply a supervised classification algorithm, Linear SVM, to distinguish the use of bad words in a casual way from invective. We also define two sets of typical and newer NLP features to analyze different aspects of the posts.

4.1 Classic Features

We make use of the following different types of lexical, syntactic, and domain related features in this case:

Lexical: We use word n-grams, char n-grams, k-skip n-grams (to capture long distance context) as features. We weigh each term with its term frequency-inverse document frequency (TF-IDF).

POS colored n-grams: We use the n-gram of tokens with their POS tags to understand the importance of the role played by the syntactic class of the token in making a post invective. We use

[6] https://www.kaggle.com/c/detecting-insults-in-social-commentary

Pattern	Example
L (You're) + R + D + A* + N (bad word)	You're just a pussy.
L (You're) + D + A* + N (bad word)	You're a one retarded b*tch.
V (bad word) + O	I want to kill(V) you(O).
O + N (bad word)	You shitheads.
N + N* (at least 2 bad words)	You stupid ass(N) dip(N) shit(N)
O (You) + A + N (bad word)	You stupid ass.
V (bad word) + D + N (bad word)	S**k my ass.

Table 4: Negative patterns for detecting nastiness. The capital letters are the abbreviations for the following POS tags: L = nominal + verbal (e.g. I'm)/verbal + nominal (e.g. let's), R = adverb, D = determiner, A = adjective, N = noun, O = pronoun (not possessive)

CMU's Part of Speech tagger[7] to get the POS tags for each document.

Emoticons (E): We use a normalized count of happy, sad and total emoticons as features to feed the classifier.

SentiWordNet (SWN): We use sentence neutrality, positive and negative scores using SentiWordNet (Baccianella et al., 2010), average count of nouns, verbs, adverbs and adjectives (Ark Tweet NLP (Owoputi et al., 2013)) as features.

LIWC (Linguistic Inquiry and Word Count): LIWC2007 (Pennebaker et al., 2007)) helps us to determine different language dimensions like the degree of positive or negative emotions, self-references, and casual words in each text. In this case, we use a normalized count of words separated by any of LIWC categories.

Style and Writing density (WR): This category focuses on the properties of the text by considering the number of words, characters, all uppercase words, exclamations, question marks, average word length, sentence length, and words per sentence as the features.

Question-Answer (QA): As we work with a data set from a semi-anonymous social network that contains question-answer pairs, certain features like type of post (question or answer), whether the post is a reply to an anonymous post, user mentions in the post, bad word ratio and bad words can be useful for detecting invective posts.

Patterns (P): Based on work by (Yin et al., 2009) and careful review of our training set, we extract the patterns (combination of lexical forms and POS tags) presented in Table 4, and define the binary feature vector to check the existence of any of them in the post.

[7] http://www.cs.cmu.edu/~ark/TweetNLP/#pos

4.2 Newer Features

In this set of features, we define the features listed below:

Embeddings: The idea behind this approach is to use a vector space model to improve lexical semantic modeling (Le and Mikolov, 2014). We use two different types of features in this case. The first one is defined by averaging the word embedding of all the words in each post, and the second one is based on a document embedding approach.

LDA: In order to find and analyze the topics involved in invective posts, we employ one of the best known topic modeling algorithms, Latent Dirichlet Allocation (LDA) (Blei et al., 2003). In this case, for each post we make a feature vector containing the probability of appearance of each topic in it.

5 Experiments and Results

In this section, we evaluate our methods on three different data sets we presented in Section 3. Our goal is to show our model works well not only for our data set, but also for the others.

5.1 Experimental Setup

For our data set, we randomly divide the data into training and test in a 70:30 training-to-test ratio, preserving the class distribution of both invective and neutral classes. We use 20% of the training data as a validation set to search for the best C parameters for the Linear SVM through grid search over different values. Since the data set is highly skewed, we perform oversampling of the invective instances during training to mitigate the imbalance data problem. Note that Kaggle corpus and Wikipedia corpus contain training, evaluation, and test sets separately.

Moreover, for the embedding features, we build the vector space by training 290,634 unique words coming from all 586K question-answer pairs we crawled from ask.fm. Also for the LDA feature, using all crawled data from ask.fm, we consider all pairs related to each user as a single document, and ignore the users with less than 10 pairs. For the other two data sets, we look at each comment as a single document. In the pre-processing step, we remove stopwords and words that occur less than 7 times, and set the number of topics to 20.

5.2 Evaluation

People use emoticons to help convey their emotions when they are posting online. In our baseline experiment, at first we simply check whether a post contains any emoticons in the list {*<3*, *:-)*, *:)*, *(-:*, *(:*, *:o)*, *:c)*} since by looking at the training data we found that these emoticons were used to show positive feelings. If the post contains at least one of these emoticons, we label it "neutral". Otherwise, we calculate the ratio of bad words to total words. If it is greater than a given threshold, our baseline system predicts the post as "invective".

$$invective(x) = \begin{cases} 0, & if\ badWordRatio(x) < T \\ 1, & if\ badWordRatio(x) \geq T \end{cases} \quad (2)$$

In this research, since we work with highly imbalanced data sets, we used "f1-score" and "area under the receiver operating characteristic curve (AUC)" as the evaluation metrics as they are less sensitive to imbalanced classes. Table 5 shows the results for our baseline experiment. We find the best threshold value among all threshold values {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8} by performing grid search using the training set for training, and the validation set for testing.

With the feature collections we discussed in section 4, we train a Linear SVM classifier. Similar to the baseline experiment, for each set of features, we tuned SVM C parameter (inverse of regularization strength) with a grid search over values {0.00001, 0.0001, 0.001, 0.01, 0.1, 1.0, 10, 100, 1000, 10000}. Table 6 shows the classification results of invective class for all the features and some of their combinations for all three different data sets. Please note that Question-Answer feature can not be applied on Kaggle and Wikipedia data, because the comments format in these data sets are not of question-answer type.

5.3 Classification Results

The last row of Table 6 shows that combining all features does not always give the best F1-score. We obtained an F1-score of 0.59 for our data when we selectively combined different types of features. Although some features like SWN and P alone perform worse or not much better than the baseline (comparing AUC or F1-score), it seems that selectively combining them with other features improves the performance of the system. We can see in the results that when we combine a feature with others, in most cases but not all we get a higher AUC score compared to only using a single feature for training the classifier. This means each feature carries valuable information about different aspects of the posts. It is very interesting that combining emotion based features with the embedding ones (LIWC+E+SWN+W2V+D2V) gives us one of the best AUC scores. It shows that the emotions reflected from the text give us good information about whether it is hostile or not. However, the results we got from LDA features are not remarkable. Even combining this feature with the others does not seem to improve performance. One reason may be the sparsity of feature vectors in this case. LDA features ranks all trained topics over each document. It makes a vector for each post containing the probability of appearing each topic in it. Since generally, the length of online comments is very short, this vector would be very sparse.

Table 6 also shows the results for the Kaggle and Wikipedia data sets. Our results do not outperform the best AUC score reported by Kaggle's winner (0.842[8]). However, we consider our method promising, since our features are not customized for Kaggle data set. Also, we compare our results with those reported for Wikipedia data (Wulczyn et al., 2016). They only presented the AUC of their different model architectures trained on the train split and evaluated on the development split. With the same configuration, we found that our results are similar to those they reported (e.g. using the same experimental set up, they got an AUC of 0.952 for word n-gram, and we got an AUC of 0.956 for word unigram). Overall, the results of applying our model on Kaggle and Wikipedia data show that it is applicable to

[8] https://www.kaggle.com/c/detecting-insults-in-social-commentary/leaderboard

	Our data set		Kaggle data set		Wikipedia data set	
Experiment	AUC	F-score	AUC	F-score	AUC	F-score
Random Baseline	0.492	0.26	0.513	0.35	0.509	0.17
Our Baseline	0.567	0.27	0.597	0.36	0.610	0.28

Table 5: Baseline experiment results for invective class

	Our data set		Kaggle data set		Wikipedia data set	
Feature	**AUC**	**F-score**	**AUC**	**F-score**	**AUC**	**F-score**
Unigram (U)	0.768	0.57	**0.813**	0.71	0.882	0.72
Bigram (B)	0.680	0.48	0.742	0.62	0.810	0.66
Trigram (T)	0.587	0.31	0.647	0.46	0.702	0.53
Word 1, 2, 3gram (UBT)	0.726	0.55	0.777	0.68	0.830	0.74
Char 3gram (CT)	0.753	0.55	0.805	0.70	0.883	0.69
Char 4gram (C4)	0.748	0.56	0.812	0.72	0.879	0.73
Char 5gram (C5)	0.717	0.52	0.793	0.71	0.869	0.74
Char 3, 4, 5gram (C345)	0.734	0.55	0.811	**0.73**	0.866	0.75
2 skip 2gram (2S2G)	0.654	0.44	0.756	0.65	0.764	0.65
2 skip 3gram (2S3G)	0.593	0.32	0.649	0.46	0.712	0.52
POS colored unigram (POSU)	0.762	0.56	0.803	0.70	0.874	0.71
POS colored bigram (POSB)	0.674	0.47	0.732	0.61	0.806	0.65
POS colored trigram (POST)	0.577	0.28	0.643	0.45	0.697	0.52
POSU+POSB+POST (POS123)	0.724	0.55	0.788	0.68	0.824	0.73
Question-Answer (QA)	0.744	0.52	N/A	N/A	N/A	N/A
Emoticon (E)	0.511	0.30	0.505	0.41	0.524	0.19
QA + E	0.743	0.52	N/A	N/A	N/A	N/A
SentiWordNet (SWN)	0.602	0.35	0.575	0.39	0.632	0.30
C345 + SWN	0.736	0.55	0.797	0.72	0.866	0.75
LIWC	0.662	0.42	0.715	0.57	0.787	0.53
QA + LIWC	0.764	0.55	N/A	N/A	N/A	N/A
Writing Density (WR)	0.564	0.30	0.566	0.42	0.682	0.31
U + WR	0.769	0.57	0.804	0.70	0.878	0.71
Patterns (P)	0.539	0.17	0.518	0.09	0.544	0.16
QA+LIWC+P	0.756	0.54	N/A	N/A	N/A	N/A
Word2vec (W2V)	0.745	0.51	0.759	0.63	0.854	0.61
Doc2vec (D2V)	0.750	0.52	0.792	0.66	0.886	0.60
LDA	0.626	0.37	0.559	0.40	0.577	0.26
LIWC+E+SWN+W2V+D2V	0.780	0.56	0.799	0.68	**0.889**	0.65
U+C4+QA+LIWC+E+SWN+W2V+D2V	**0.785**	0.57	N/A	N/A	N/A	N/A
U+C4+POSU+QA+D2V+LDA	0.781	0.58	N/A	N/A	N/A	N/A
C4+U+QA+E	0.766	**0.59**	N/A	N/A	N/A	N/A
All Features	0.756	0.56	0.798	0.71	0.882	**0.75**
Best Previous Reported score	_	_	0.842	_	_	_

Table 6: Classification results for invective class. N/A stands for the features that are not applicable to Kaggle and Wikipedia data sets

other data sets. According to the comparison of all three corpora in Section 3.3, we believe that the major reasons why we get higher scores in those two other data sets comparing with ours are:

1. In ask.fm, comments are question-answer pairs which are shorter than in other data sets. By looking at our data, we found that in many cases both question and answer include only one word – that makes the decision hard.

2. Online posts do not basically follow formal language conventions. Since ask.fm is mostly used by teenagers and youth, there are more misspellings and abbreviations inside the texts, which makes their processing much more difficult.

Among all the features, only P works poorly specifically in Kaggle data. But as mentioned in Section 4, for extracting those patterns, we only looked at our training data. So, it is understandable that they may not give us good results for the other data sets. So, it would be interesting to find a way for extracting the negative patterns from the text automatically.

Table 7 lists important features learned by the classifier. The "_" represents the whitespace char-

Feature	Our data set	Kaggle data set	Wikipedia data set
U	bitch, fuck, asshole, shut, stfu, off, you,stupid, fucking, ugly, pussy, u, ass, slut, face	you, idiot, stupid, dumb, loser, your, moron, ignorant, you're, faggot, bitch, shut, asshole, ass, retard	fuck, fucking, stupid, idiot, shit, asshole, ass, moron,bullshit, suck, idiots, bitch, sucks, dick, penis
C4	itch, bitc, _ass, _fuc, uck_, stfu, _hoe, _bit, tfu_, fuck, _stf, dumb, _off, _you, slut,	_you, you_, re a, diot, _idi, idio, dumb, moro, oron, _dum, your, bitc, tard, _fuc, oser	fuck, _fuc, shit, uck_, diot, _ass, suck, idio, moro, _shi, _gay, bitc, oron, dick

Table 7: Top negative features

acter. It is good to see that the classifier has learned to discriminate between neutral and invective words. The most interesting point obtained from this table is that the second-person pronoun is ranked as one of the top negative features. It supports our idea that invective posts have specific patterns in most of the cases. Also, in our data set, the word "face" ranked as a highly negative feature. It shows that attackers post negative comments about victims' faces, and in some cases as an answer to an uploaded picture. Moreover, the bad words captured from the other data sets (like *idiot*, *stupid*, *moron*) give us some idea to expand our bad word lists to enrich our data set.

	Posts
1	***Answer**: stfu* ***Answer**: Die*
2	***Question**: Fuck you brian lmao* ***Answer**: xD ty*
3	***Question**: Can I kill you?* ***Question**: Can we fuck please?*
4	***Question**: You are hot as fuck*

Table 8: Examples of misslabeled instances by the classifier

Analyzing mistakes, we found that the classifier gets confused with single profane word answers (Example 1 in Table 8), question and answer pairs in which users joke around using profanities (Example 2 in Table 8), posts with mixture of politeness and profanity (Example 3 in Table 8), and posts with bad words that are offered as compliments (Example 4 in Table 8).

5.4 Negativity of words

Table 9 shows the degree of negativity for the words in our bad-word list. We do this analysis in order to identify how negative each word in our bad word list is by itself. For computing this measure, we consider the posts that contain only one profane word. Then, for each bad word w_i in the list, we apply the same formula as Equation 1 to calculate the ratio of the negative posts containing w_i or any of its variations to the total posts in which w_i or any of its variations appears as their single bad word.

bad word	negativity	bad word	negativity
as**ole	51.16%	b*tch	41.65%
kill	12.47%	a*s	24.77%
f*ck	33.05%	die	7.41%
n**ger	13.30%	s*ck	26.88%
sh*t	15.23%	h*e	36.58%
cut	4.85%	stfu	51.55%

Table 9: Degree of negativity for bad words

From Table 9, it is clear that most of our bad words are used in a neutral or positive way more often than in a negative way. Although these numbers are related also to the overall incidence of nastiness, there are some noteworthy findings. For example, the word "f*ck", when used as a verb, referring to sexual activity, was used more often in a neutral or positive post, rather than a negative post. Thus its overall negative score is 33.05% compared to the word "as**ole" that had a negative score of 51.16%. This finding reflects a sexualized teen culture, part of a growing problem affecting young social media users. The low degree of negativity of the words "die", "kill", and "cut" are also interesting. By looking at the data, we find that the likelihood that these harm-related words reflect an online harassment is related to the appearance of the other bad words. Moreover, the data shows that these words are used sometimes to threaten people or encourage them to commit suicide. In contrast, the acronym "stfu" has the strongest degree of negativity. We believe that these observations are related to the versatility of the words. It is less likely to see the acronym "stfu" being used in a neutral and positive way than the other words. Also, some words like "suck" and "hoe" seem to carry a highly negative weight.

6 Conclusion and Future Work

In this paper, we present our evolving approaches for creating a linguistic resource to investigate nastiness in social media. We start out by selecting profanity-laden posts as a likely hostile starting point for invective potentially leading to cyberbullying events. We use various types of classic and new features, and try to combine them for distinguishing extremely negative text from the rest. Also, by applying our machine learning model on Kaggle and Wikipedia data, we show that this model can be applicable to other data sets. Interestingly, we find that profanities and vulgarities abound in teens posts and that the degree of negativity of profanities varies, from the strong negativity of the acronym "stfu" to the ambiguity of the term "f**k" which when used as a verb referring to sexual desire or propositioning is sometimes considered a compliment. We find interesting trends, degrees of negativity in profanity that possibly indicate heavy use of profanity among teens, and also reflect a sexualized teen culture.

We are continually enriching this linguistic resource by identifying more types of abusive posts. Future plans for our research are to capture more emotional aspects from the online comments, extract negative patterns from the text automatically, and consider a topic modeling algorithm specifically designed for short texts in order to extract only one topic per document. We also plan to work on a graph model of the users to better identify cyberbullying episodes.

Acknowledgments

We thank the anonymous reviewers for their valuable feedback on this research.

References

Haider M. al-Khateeb and Gregory Epiphaniou. 2016. How technology can mitigate and counteract cyberstalking and online grooming. *Computer Fraud & Security* 2016(1):14–18.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*. volume 10, pages 2200–2204.

David M. Blei, Andrew Y. Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3(Jan):993–1022.

Karthik Dinakar, Birago Jones, Catherine Havasi, Henry Lieberman, and Rosalind Picard. 2012. Common sense reasoning for detection, prevention, and mitigation of cyberbullying. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 2(3):18.

Healy. 2014. Ask.fm is relocating to ireland and no one is happy about it. http://mashable.com/2014/11/05/ask-fm-relocation-ireland-cyberbullying-suicides-cold-shoulder/#SdafIlqyoGqg.

Homa Hosseinmardi, Rick Han, Qin Lv, Shivakant Mishra, and Amir Ghasemianlangroodi. 2014. Towards understanding cyberbullying behavior in a semi-anonymous social network. In *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*. IEEE, pages 244–252.

Krishna B. Kansara and Narendra M. Shekokar. 2015. A framework for cyberbullying detection in social network. *International Journal of Current Engineering and Technology* 5(1).

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*. volume 14, pages 1188–1196.

Henry Lieberman, Karthik Dinakar, and Birago Jones. 2011. Let's gang up on cyberbullying. *Computer* 44(9):93–96.

Sonia Livingstone, Leslie Haddon, Anke Görzig, and Kjartan Ólafsson. 2010. Risks and safety on the internet. *The Perspective of European Children. Final Findings from the EU Kids Online Survey of* pages 9–16.

Jamie Macbeth, Hanna Adeyema, Henry Lieberman, and Christopher Fry. 2013. Script-based story matching for cyberbullying prevention. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*. ACM, New York, NY, USA, CHI EA '13, pages 901–906. https://doi.org/10.1145/2468356.2468517.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, WWW '16, pages 145–153. https://doi.org/10.1145/2872427.2883062.

nobullying.com. 2015. The complicated web of teen lives - 2015 bullying report—nobullying—. http://nobullying.com/the-complicated-web-of-teen-lives-2015-bullying-report/.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. Association for Computational Linguistics.

Justin W. Patchin. 2015. Summary of our cyberbullying research (2004-2015). http://cyberbullying.org/summary-of-our-cyberbullying-research.

Justin W. Patchin and Sameer Hinduja. 2010. Cyberbullying and self-esteem. *Journal of School Health* 80(12):614–621.

James W. Pennebaker, Roger J. Booth, and Martha E. Francis. 2007. Liwc2007: Linguistic inquiry and word count. *Austin, Texas: liwc.net* .

Shute. 2013. Cyberbullying suicides: what will it take to have ask.fm shut down? - telegraph. http://www.telegraph.co.uk/news/health/children/10225846/Cyberbullying-suicides-What-will-it-take-to-have-Ask.fm-shut-down.html.

Fabio Sticca and Sonja Perren. 2013. Is cyberbullying worse than traditional bullying? Examining the differential roles of medium, publicity, and anonymity for the perceived severity of bullying. *Journal of youth and adolescence* 42(5):739–750.

Cynthia Van Hee, Els Lefever, Ben Verhoeven, Julie Mennes, Bart Desmet, Guy De Pauw, Walter Daelemans, and Veronique Hoste. 2015. Detection and fine-grained classification of cyberbullying events. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*. INCOMA Ltd. Shoumen, Bulgaria, pages 672–680. http://aclweb.org/anthology/R15-1086.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2016. Ex machina: Personal attacks seen at scale. *CoRR* abs/1610.08914. http://arxiv.org/abs/1610.08914.

Jun-Ming Xu, Kwang-Sung Jun, Xiaojin Zhu, and Amy Bellmore. 2012. Learning from bullying traces in social media. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Stroudsburg, PA, USA, NAACL HLT '12, pages 656–666. http://dl.acm.org/citation.cfm?id=2382029.2382139.

Dawei Yin, Brian D. Davison, Zhenzhen Xue, Liangjie Hong, April Kontostathis, and Lynne Edwards. 2009. Detection of harassment on web 2.0. *Proceedings of the Content Analysis in the Web 2.0 (CAW2.0)* 2:1–7.