Detection violent and abusive content in social media

Problem Statement

Every day, billions of people communicate on the online social network. Facebook, with more than a billion of users, is currently the largest and most popular OSN in the world. Other known OSNs are Google+, with over 235 million of users; Twitter, with more than 200 million users; and LinkedIn, with around 160 million users (Fire M, Goldschmidt R, Elovici Y. et al, 2014). Usage of social network is growing rapidly for sharing private and/or intimate information by various applications that assist users to get in close contact with others without considering cyber security breaches. These kinds of communication may lead towards some hazardous outcomes in terms of injecting various kinds of security attacks in social network. Message posts can contain the sharing of some kinds of abusive or offensive contents which can emerge the threats like cyberbullying. Usually, Adults can be able to put a line of secure communication and are having better awareness of often curious to explore new fields without the ability the existing dangers in social networks bring along. By contrast, children or teenagers often have wrong threat perception and are to weigh up potential risks (Rybnicek M, Poisel R, Tjoa S, et al, 2013).

Anytime one engages online, whether on message board forums, comments, or social media, there is always a serious risk that he or she may be the target of ridicule and even harassment. Words and sentences such as kill yrslef a$$hole or they should all burn in hell for what they've done are unfortunately not uncommon online and can have a profound impact on the civility of a community or a user's experience. To combat abusive language, many internet companies have standards and guidelines that users must adhere to and employ human editors, in conjunction with systems which use regular expressions and blacklist, to catch bad language and thus remove a post. As people increasingly communicate online, the need for high quality automated abusive language classifiers becomes much more profound.

Background

Detecting abusive language is often more difficult than one expects for a variety of reasons. the noisiness of the data in conjunction with a need for world knowledge not only makes this a challenging task to automate but also potentially a difficult task for people as well.

More than simple keyword spotting. The intentional obfuscation of words and phrases to evade manual or automatic checking often makes detection difficult. Obfuscations such as ni9 9er, whoopiuglyniggerratgolberg and JOOZ make it impossible for simple keyword spotting metrics to be successful, especially as there are many permutations to a source word or phrase. Conversely, the use of keyword spotting could lead to false positives.

Difficult to track all racial and minority insults. One can make a reasonably effective abuse or profanity classifier with a blacklist (a collection of words known to be hateful or insulting), however, these lists are not static and are ever changing. So a blacklist would have to be regularly updated to keep up with language change. In addition, some insults which might be unacceptable to one group may be totally fine to another group, and thus the context of the blacklist word is all important

Abusiveness can be cross sentence boundaries. In the sentence Chuck Hagel will shield Americans from the desert animals bickering. Let them kill each other, good riddance!, the second sentence which actually has the most hateful intensity (them kill each other) is dependent on the successful resolution of them to desert animals which itself requires world

knowledge to resolve. The point here is that abusive language is not limited to just the sentence. In some cases, one has to take the other sentences into account to decide whether the text is abusive or carries incidences of hate speech.

Sarcasm. Finally, we noted cases where some users would post sarcastic comments in the same voice as the people that were producing abusive language. This is a very difficult for humans or machines to get correct as it requires knowledge of the community and potentially even the users themselves: same thing over and over and over and over day in night and day 'cause i am handicapped and stay home. i hate jews they ran over my legs with their bmw. so i will blast them everyday. I really hurt them i am so powerful .. If ipost about jews here they all suffer. im sow powerfull bwbwbwbwaaahahahahahahah im a cripple but i can destroy them with my posts.. I am super poster. Bwwbwahahahaha noone can find me .. I am chicken so i can post behind yahoos wall of anonymous posters. Bwbwbwbabahahahah i will give him ten thumbs down and slander jews.. Bwbwbwbahahahah..i am adoplh hitler reincarnated.

Methodology

We want to employ a supervised classification method which uses NLP features which measure different aspects of the user comment. Specifically, we use the Vowpal Wabbit's regression model 5 in its standard setting with a bit rate of 28. We base our NLP features on prior work in sentiment, text normalization among others. Our features can be divided into four classes: N-grams, Linguistic, Syntactic and Distributional Semantics. For the first three features, we do some mild pre-processing to transform some of the noise found in the data which could impact the number of sparse features in the model. Example transformations include normalizing numbers, replacing very long unknown words with the same token, replacing repeated punctuation with the same token, etc. For the fourth feature class, we did none of the above normalization.

N-gram Features: We employ character n-grams (from 3 to 5 characters, spaces included) and token unigrams and bigrams. In contrast to prior work in this field which either ignored unnormalized text or used simple edit distance metrics to normalize them, we use character n-grams to model the types of conscious or unconscious bastardizations of offensive words.

Linguistic Features: To further handle the noisiness of data, we developed specialized features based on work by. These features are intended to explicitly look for inflammatory words (such as the use of pre-existing hate lists) but also elements of non-abusive language such as the use of politeness words or modal verbs. These features include:

• length of comment in tokens

• average length of word

• number of punctuations

• number of periods, question marks, quotes, and repeated

Syntactic Features: The use of natural language parsing is common for tasks ranging from sentiment analysis to best answer prediction in CQA analysis. We derive features from the ClearNLP v2.0 dependency parser7 . The features are essentially different types of tuples making use of the words, POS tags and dependency relations. These include: • parent of node • grandparent of node • POS of parent • POS of grandparent • tuple consisting of the word, parent and grandparent • children of node8

Experimental Design

In this section, we describe a battery of experiments meant to evaluate our classifier, compare it to prior work and then use it as a tool to analyze trends of hate speech in user comments. Here we show the overall performance of our model on the Primary Finance and News data sets. We evaluate the impact of each feature and discuss which are best for this task.

We are then comparing our model to the prior work on the WWW2015 set. Next, we evaluate on our curated Evaluation data set (§5.3) and in §5.4 we investigate the question: How does performance vary over time? One could hypothesize that language and use of hate speech changes rapidly and this will thus impact a classifier's performance if the model is not updated.

Evaluation on Primary Data Set. In this set of experiments, we train and test our model using the Primary Data Set for both domains (Finance and News). For each domain, we use 80% for training and 20% for testing. Table 1 shows the results for each domain when a model trained with a single feature type as well as with all features combined. For both domains, combining all features yields the best performance (0.795 for Finance and 0.817 for News). News has a slight performance edge though that may be easily accounted for by the fact that there is a larger training corpus available for that domain. In terms of individual features, for both sets, character ngrams have the largest contribution. The two sets do exhibit different behaviour in terms of other features. In the Finance set, the syntactic and distributional semantics features do not perform as well as they fare in the News domain. We believe that the Finance domain is slightly noisier than News and thus these more complex features do not fare as well.

Features	**Finance**	**News**
Lexicon	0.539	0.522
Trained Lexicon	0.656	0.669
Linguistic	0.558	0.601
Token N-grams	0.722	0.74
Character N-grams	0.726	0.769
Syntactic	0.689	0.78
word2vec	0.653	0.698
pretrained	0.602	0.649
comment2vec	0.68	0.758
All Features	0.795	0.817

Table:1 Primary Data Set Results (by F-score)

Evaluation on Temporal Data Set: For our final set of experiments, we seek to answer the following questions: 1) how much training data is actually necessary for a high performance? and 2) does performance degrade over time if a model is not updated? To answer these questions we ran three experiments using the Temporal Set (Data Set 2) which is divided into consecutive slices of 20k comment each.

1. Original We use the model developed using Primary Data Set and used in the evaluation §5.1, and evaluate it over the consecutive slices of data in the Temporal Set. Our hypothesis

is that if there is significant language change in user comments, performance should degrade. This would mean that any anti-abuse method would need to be updated regularly.

2. Each Slice We train a model with the data at each slice (t) and apply the model to the next slice (t + 1). So each training set consists of only 20k comments and is markedly smaller than the other two evaluations.

3. Accumulated We train a model by accumulating data available until that the time (1..t) and apply the model to the next slice (t + 1). Our hypothesis is that this model should outperform the Each Slice model since it consists of more data, but the data is smaller than the set used in Original.