Machine Learning Approach for Detection of Cyber-Aggressive Comments by Peers on Social Media Network

Vikas S Chavan
Department of Information Science and Engineering
P.E.S Institute of Technology, Bangalore
Visvesvaraya Technological University, India
chavanvikas57@gmail.com

Shylaja S S
Department of Information Science and Engineering
P.E.S Institute of Technology, Bangalore
Visvesvaraya Technological University, India
shylaja.sharath@pes.edu

Abstract—**The fast growing use of social networking sites among the teens have made them vulnerable to get exposed to bullying. Cyberbullying is the use of computers and mobiles for bullying activities. Comments containing abusive words effect psychology of teens and demoralizes them. In this paper we have devised methods to detect cyberbullying using supervised learning techniques. We present two new hypotheses for feature extraction to detect offensive comments directed towards peers which are perceived more negatively and result in cyberbullying. Our initial experiments show that using features from our hypotheses in addition to traditional feature extraction techniques like TF-IDF and N-gram increases the accuracy of the system.**

Keywords—cyber-aggressive; supervised; machine learning;

I. INTRODUCTION

Cyberbullying is a kind of online harassment, which can be defined as rude, insulting, offensive, teasing, demoralizing comments through online social media targeting one's educational qualifications, gender, family and personal habits. According to 'Tweens, Teens and Technology 2014 Report' by McAfee[7], 50% of Indian Youth have had some experience with cyberbullying. According to a survey [10], it has been identified that a significant number of suicides have been committed by teens who were exposed to cyberbullying. Teens feel demoralized and get frustrated when they encounter such cyber-aggressive comments which act as a barrier for participation and socializing. Most networking sites today prohibit the use of offensive and insulting comments. But this partially being carried out and filtered to a limited extent. As there is enormous amount of data available it is impossible to take help of human moderators to manually flag each insulting and offensive comments. Thus, a automatic classifiers that is fast and effective to detect such type of comments is required which will further reduce cyberbullying. However, there are enormous challenges involved as comments contains many special characters eg:"Your a retard go post your head up your #%&*","U r !diot" containing insults and also some sarcastic comments. In this paper we use machine learning techniques to detect the insults and offensiveness of the comments present in social networking sites. The datasets used for experiments are collected from kaggle website [8]. The training datasets contain just 4000 comments. The model is applied on the test set which contains close to 2500 comments. The first objective is to predict whether a comment is an insult to a participant of a conversations. We have proposed two new hypotheses for detecting cyberbullying. Further a comparison between the performances of popular machine learning classification algorithms is presented.

This problem is a binary classification problem where we are trying to classify comments as bullying and non-bullying. We have identified features which detect offensive comments directed towards peers in addition to standard features extraction techniques such as TF-IDF score, N-grams, bad word count and stemming to model Supervised machine learning algorithms like Support vector machines and Logistic regression. The feature vector built using proposed features effectively detects the comments directed towards peers as bullied.

The rest of the paper is organized as follows. Section II describes the related work carried out in this field. In section III we describe the proposed method. Section IV contains the results. Finally in section V we present the future scope.

II. RELATED WORK

In an effort to model the cyberbullying, Kelly Reynolds and April Kontosthatis, 2011[1] used machine learning to train the data collected from FromSpring.me, a social networking site, the data was labeled using Amazon Web service called Turk. The number of bad words were used as a feature to train model. In a study by Dinakar et al [2], states that individual topic-sensitive classifiers are more effective to detect cyberbullying. They experimented on a large corpus of comments collected from Youtube.com website. Ellen Spertus [3] tried to detect the insult present in comments, they used static dictionary approach and defined some patterns on socio-linguistic observation to build feature vector which had a disadvantage of high false positive rate and low coverage rate. Altaf Mahmud et al [4] tried to differentiate between factual and insult statements by parsing

978-1-4799-8792-4/15/$31.00 ©2015 IEEE

comments using semantic rules, but they did not concentrate on comments directed towards participants and non-participants. Another work by Razavi et al [5] used a static dictionary and three level classification approach using bag-of-words features, which involved use of dictionary that is not easily available.

All these methods lack generality due to flexible use in conversation and uses rule-based recognition which is difficult to model. These work do not distinguish between the offensive comments directed towards the people participating in blog/forum conversation and non-participants such as celebrities, public figures etc.

We aim at building an efficient classifier on proposed features to detect cyberbullying comments directed towards peers participating in conversations over social media.

III. Proposed method

This section proposes the methodology and framework used for classification of comments. Diagrammatically it is shown in fig 1. The steps involved are Normalization, standard Feature extraction, additional feature extraction, feature selection and finally classification.

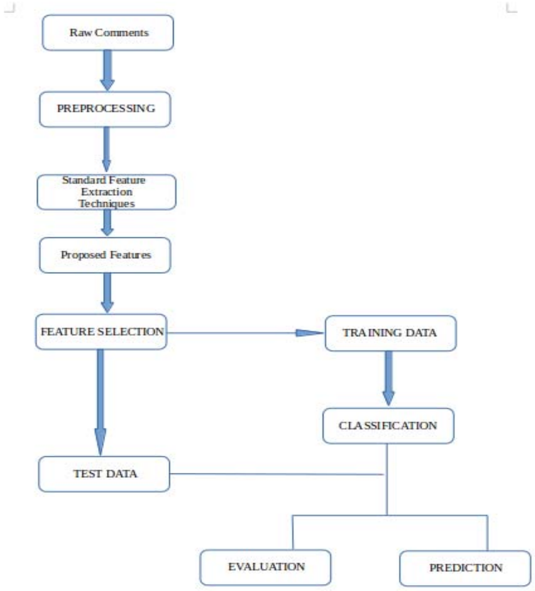
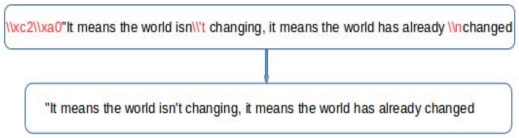


Fig. 1: Proposed Method Framework

A. Normalization

The Data set we have used contains list of comments and respective labels. These should be converted into feature vector which are used by our machine-learning algorithms. For this we use different Natural language processing techniques to obtain an accurate representation of the comments in feature vector form. We use various techniques based on our observations.

- Removing unwanted strings: For the comments to be used by machine-learning algorithms they should be in standard form. Raw comments present in dataset which contains many unwanted strings like '\xa0','\\n' and many such encoding parts should be removed. Hence the first step is to preprocess the comments by removing unwanted strings,hyphens and punctuations. The following figure demonstrates an example of this step.

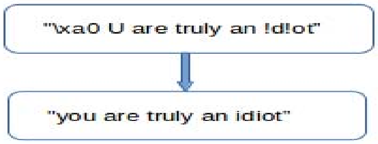


\\xc2\\xa0"It means the world isn\\'t changing, it means the world has already \\nchanged

"It means the world isn't changing, it means the world has already changed

Fig. 2: Removing unwanted strings

- Correcting words: One of the reasons comments are classified as insulting is the presence of profane or abusive words. The total number of bad words present in comments is taken as one of the features. A dictionary of 500 bad words [9] is compiled, which also includes variations of words (@$$,s h i t). This dictionary is used because people using the online forums sometimes use special characters to build a insulting word (!d!ot,@$$ole). When we encounter such words, the dictionary helps to convert them into natural form. Also, Stemming is applied to capture bad word variations that are not contained in dictionary. Stemming reduces a word to its core root, for example embarrassing is reduced to embarrass. Here it is noted that stemming is only applied to bad word dictionary not on the dataset used, as it will lead to information loss. Again a small dictionary and a spell checker is used to convert all variations of "you", "you're" (e.g u, ur etc) which are present in the dataset as participant use them as part of flexible language. Following figure demonstrates an example of this step.



"\xa0 U are truly an !d!ot"

"you are truly an idiot"

Fig. 3: Correcting words

B. Standard Feature Extraction

To train machine learning algorithms, strings should be converted in feature vector. We use N-gram, counting and TF-IDF score to construct feature vector. The process occurs in following steps. Table 1 gives a brief description of the features being used.

- N-gram model: N-grams are a group of continuous sequence of n-items from a given text. These are used for dividing text and words into n chunks known as N-grams. Consider sentence "You are funny" its unigram will be "you","are","funny". Bigram-"you are","are funny". Trigram-"funny you are", "are funny you". We use 2, 3, 4 and 5 N-grams for the building feature vector.

- Counting: Count the number of times each of these tokens occurs in each of the text strings. This way we construct a sparse matrix of size N by V where N is the size of the training data which is number of comments and V is the size of the vocabulary, the length of feature vector constructed over the whole training set using n-grams, skip grams and use of pronouns representing all the text strings where the number of occurrences of each token is a feature for that text string.
- TF-IDF Score: TF-IDF stands for "Term Frequency, Inverse Document Frequency". It is a way to evaluate the importance of words (or "terms") in a document based on how frequently they appear across various documents. The score signifies the importance of that term in relation to the original training data.

  TF-IDF score is given by:

$$\text{TF-IDF} = \text{tf}_{ij} * \text{idf}_i$$

Numerically, term frequency tfij specify the importance of a word *i* in comment *j*. It is determined as:

$$tf_{ij} = \frac{N_{ij}}{\Sigma N_j}$$

Where N_{ij} is the frequency of word *i* in comment *j* and ΣN_i is the frequency of all words in comment *j*.

Inverse document frequency idf_i specifies the importance of a word *i* in the entire training dataset. It is determined as:

$$idf_i = \frac{\log |C|}{|C_j : W_i \epsilon C_j|}$$

Where |C| is the total number of comments, $|C_j : W_i \epsilon C_j|$ is the number of comments where word W_i appears. So each comment contains a vector of words and each word is denoted in the vector by its TF-IDF score.

C. Additional Features

- Capturing pronouns: It is been observed that cyber-aggressive comments which are directed towards peers are perceived more negatively and results in cyberbullying [11]. Comments containing a pronoun like 'you' followed by a insulting or profane words are peer directed comments which are taken as negative and teens get frustrated after encountering such comments. So, to detect such comments we have used the count of pronouns as one of the features for detecting cyberbullying. To extract this feature we calculate TF-IDF score of pronoun present in comment. This feature is our strong hypothesis which greatly increases the accuracy and helps in detecting cyber-aggressive comments.
- Skip- grams: We also used skip-grams in building a feature vector as they help in detecting insult more effectively. These consider the long distance words as a feature. For example consider "You are an idiot" as a comment, if we use 2-skip-gram, count of 'You are' as one feature and 'an idiot' as other is added in our feature-matrix. This way, the comments containing co-occurrences of words like "You idiots" which is negative and will be detected using skip-grams.

TABLE I. FEATURES SETS

Features	Description
N-gram	Used unigram, bigram and trigram as binary features
Count	Tokenized the comments and count the occurrence of each token in it. This way we created a sparse matrix of NxV.
TF-IDF score	Used to calculate the importance of words in documents based on how frequently they are used
Occurrence of pronouns	This is additional feature which helps in detecting cyber-aggressive comments based on pronoun "You".
Skip-grams	Adds a long distance words as a feature. Used to detect co-occurrences of some words like "You idiot".

D. Feature Selection

The machine learning algorithms cannot handle all the features which are order of some hundred thousand. So we need to select best features out of our set of features. We use a statistical hypotheses method known as "Chi Squared test" to our feature matrix to select k best features where k is parameter roughly equal to 3000.

1. Chi-Square Method: chi square (X^2) method is commonly used for selecting best features. This metric calculates the cost of a feature using the value of the chi-squared statistics with respect to class. Initially, a hypothesis H0 is assumed that the two features are unrelated, and it is The initial hypothesis H0 is the assumption that the two features are unrelated, and it is tested by chi squared formula as is shown in equation (1)

$$X^2 = \frac{\Sigma(O_{ij} - E_{ij})}{E_{ij}} \qquad (1)$$

Where O_{ij} is the observed frequency and E_{ij} is the expected frequency, asserted by the null hypothesis. Higher the value of (X^2), greater the evidence against the hypothesis H0, hence more related is the two variables. Lesser the value of (X^2), the hypotheses tends to be true, the variables are independent.

2. To understand this measure better consider the following example.

Text	'You'	'the'	'you are'	'idiot'	Label
1	Present	Present	Not-present	Present	offensive
2	Present	Present	Not-present	Not-present	Non-offensive
3	Not present	Not-present	Present	Present	offensive

Considering 'you' and 'idiot' be both independent, then expected number of rows where these happen to be present is given by

$$E('you\ present', offensive) = \frac{N('you\ present') * N('offensive')}{N}$$

Where N ('you present', offensive) is the number of rows which have the feature 'you' and are labeled as offensive and N is total number of rows.

$$X^2 = \frac{\Sigma(observed\ (i,j) - Expected(i,j))^2}{Expected(i,j))}$$

Where i={'yes present', 'yes not present'} and j ={'offensive' ,'non-offensive'}. Higher these values more related are two variables.

E. Classification

Once the features are built, we extract the best features using chi-squared test and apply the machine learning algorithms to train models on it. We have used SVM and logistic regression on our feature data. A brief summary of these algorithms are given below.

- Support vector machine (SVM): This algorithm maps the training data into feature space using kernel functions and then separates the dataset using large hyperplane. We have used linear kernel function.

$$K(x_i, x_j) = x_j x_i^T$$

Where $K(x_i, x_j)$ represents dot product of input data points x_i mapped into large dimensional feature space x_j by transformation function.

- Logistic Regression: This algorithms provides probabilistic approach to data. The outcome are probabilities modeled as a function of predicted variables, using a logistic function given below.

$$P_i = \frac{1}{(1+\ e^{-})}$$

Where P_i is the probability at observation *i.* Here Θ is calculated as:

$$\theta = \sum_{j=0}^{M} \beta_j X_{ij}$$

For $i = 1 \dots . N(no. of\ observations)$ and $j = 1 \dots . M(no. of\ independent\ variables), x_{ij}$ is j^{th} variable at observation *i.* β_j is the regression coefficient.

We get the final results by combining the results obtained by both algorithm. The final output is the probability of comment being insulting. The test dataset which is classified contains 2647 comments.

IV. Results

The model is tested over the test dataset which contains 2647 comments. Out of these, 720 comments are negative comments.

A. Datasets

The datasets we use for our experiments are taken from Kaggle[7] Website- an online competition site. The data consists of a label column followed by two attribute fields namely timestamps and Unicode-escaped text of English language comment. The datasets contains training and test datasets.

B. Evaluation parameters

Following parameters are used to compare the individual algorithms on the test datasets.

1. Recall: The recall is the ratio tp / (tp + fn) where tp is the number of true positives and fn is the number of false negatives. It is the ability of the classifier to find all the positive samples. From Table IV we can observe that using features obtained using our hypothesis increases the recall value. This show that comments which are true bullied are predicted as bullied.
2. Precision: The precision is the ratio tp / (tp + fp) where tp is the number of true positives and fp is the number of false positives. It is the ability of the classifier not to label a sample as positive that is negative.
3. AUC (Area under the curve score): Computes the Area Under the Curve (AUC) from prediction scores and this evaluation parameter is strictly restricted to binary classification. As our task is to classify the comments as a bully/not bully i.e. a binary classification task, this evaluation parameter is very important. Using traditional feature extraction technique, model resulted in AUC score of 82%. An increase of 4% is achieved after introducing features extracted from our hypotheses.
4. ACC score: In multilabel classification, this function computes subset accuracy i.e. the set of labels predicted for a sample must exactly match

the corresponding set of labels in ground truth (correct) labels.

C. Analysis

At first we build feature vector containing standard feature extraction containing TF-IDF and N-grams. Then we train our algorithms based on these feature vector and the best accuracy achieved is of logistic regression of 83%. Then, we include occurrence of pronouns and skip-gram as features which increased the accuracy and logistic regression outperformed in this too with 86%. The test datasets used for our experiment contained nearly 3000 unlabeled comments. Also, we tried to train the system with all features using SVM and logistic regression. An experimental result shown in Table II suggests that comments targeted towards peers helps in detecting cyberbullying more efficiently. Table III shows the performance of algorithms trained on the standard features extraction techniques. Table IV shows the accuracy (AUC score), precision and recall values after introducing skip-grams and pronouns as features. Following figure 4 shows the increase in accuracy by introducing additional features in addition to traditional features.

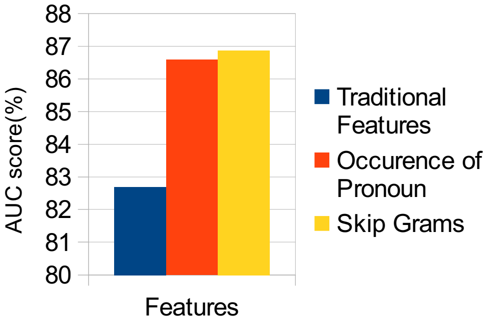


Fig. 4: Increase in AUC score by adding additional features

TABLE II. COMPARISON OF USING DIFFERENT FEATURES

Features	AUC Score
Standard features extraction	82.69
Occurrence of pronouns	86.58
Skip grams	86.87

TABLE III. PERFORMANCE OF ALGORITHMS ON TEST DATA USING STANDARD FEATURES

Algorithm	ACC score	Recall	Precision
Logistic regression	73.76	0.6147	0.644
SVM	77.65	0.5829	0.7029

TABLE IV. PERFORMANCE OF MODEL WITH DIFFERENT SKIP- GRAMS

Skips	*AUC score*	*Recall*	*Precision*
2 skips	86.84	0.72	0.765
3 skips	86.84	0.71	64.64
2,3 skips	86.92	0.71	0.769

V. CONCLUSION AND FUTURE WORK

In this paper, we presented two new hypothesis for feature extraction which can be helpful in detecting cyberbullying. We built a model which predicted comments as bully/non-bully. The end result is probability of comment being offensive to participants. Results show that our hypothesis increases the accuracy by 4% and can be used to detect the comments that are targeted towards peers.

Future work should be directed towards detecting sarcastic comments.

REFERENCES

[1] Reynolds, K.; Kontostathis, A.; Edwards, L., "Using Machine Learning to Detect Cyberbullying," Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on, vol.2, no.,pp.241,244,18-21Dec.2011.

[2] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the Detection of Textual Cyberbullying," in Proc. IEEE International Fifth International AAAI Conference on Weblogs and Social Media, Barcelona, Spain, 2011.

[3] Spertus, E., Smokey: Automatic recognition of hostile messages. In: Proceedings of the Ninth Conference on Innovative Applications of Artificial Intelligence, pp. 1058–1065 (1997)

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

[5] Razavi, A.H., Inkpen, D., Uritsky, S., Matwin, S., Offensive language detection using multi- level classification. In: Proceedings of the 23rd Canadian Conference on Artificial Intelligence, pp. 16–27 (2010)

[6] Xiang, G., Hong, J., & Rosé, C. P. , Detecting Offensive Tweets via Topical Feature Discovery over a Large Scale Twitter Corpus, Proceedings of The 21st ACM Conference on Information and Knowledge Management, Sheraton, Maui Hawaii, October 29–November 2, (2012).

[7] McAfee. (2014). *Tweens, Teens and Technology 2014.*

[8] For dataset:

Available: www.kaggle.com

[9] For bad words file:

Available:http://urbanoalvarez.es/blog/2008/04/04/bad-words-list/

[10] Hinduja, S.; Patchin, J. W. (2009). Bullying beyond the schoolyard: Preventing and responding to cyberbullying. Thousand Oaks, CA: Corwin Press. ISBN 1412966892.

[11] Elizabeth Whittaker & Robin M. Kowalski (2015) Cyberbullying via Via Social Media, Journal of School Violence, 14:1, 11-29