Detecting Different Categories of Cyber-Aggression

Zenefa Rahaman[1], **Autumn Slaughter**[2], **Elena Newman**[2], **Sandip Sen**[1],
[1] Tandy School of Computer Science, The University of Tulsa
[2] Department of Psychology, The University of Tulsa
zenefa-rahaman@utulsa.edu, sandip-sen@utulsa.edu

Abstract

Cyber-harassment is an alarming issue with widespread use of social media. These online communication and information sharing platforms not only empower people to express their views and share their opinions but also reveal an abundance of unfortunate intimidatory and hateful aggression towards individuals. Cyber harassment is an aggressive and unwanted online behavior that thrives on its intimidation of a victim. The behavior is typically frequent and repeated over time, but can also occur as an isolated incident. These nasty and often coordinated victimization of individuals have significant social costs ranging from social ostracism to opinion marginalization and suppression can have individual health costs ranging from anxiety and depression to severe outcomes such as suicide ideation. The recent study [Duggan, 2014] by the Pew Research Center found 40 percent of adult Internet users have experienced harassment online, with young women enduring particularly severe forms of it. A number of computational studies have developed automated mechanisms for detection of unwarranted victimization and harassing attacks on social network and microblogging platforms. We believe, however, that there is a compelling need for a more comprehensive suite of detection and intervention mechanisms that is grounded in a well-founded theory of human aggressive and predatory behavior. In this paper, we propose some new mechanisms detects different categories of harassment that appears on social media and compare them with other existing techniques.

1 Introduction

The rapid development of online communication and information sharingplatforms and the enthusiastic participation of the newly empowered citizens have enabled peer-to-peer communication at unprecedented scale and diversity. Social media engagement has become an issue of increasing social, economic and political importance. Individuals freely participate and express their views and opinions on diverse topics at times and from locations at their convenience. Millions of posts, in the form of texts, images, and videos, appear daily on popular websites and social media such as Facebook, Instagram, Twitter, YouTube, Tumblr, etc. Authors of those posts write about their life, share opinions on a variety of topics and discuss current issues. Hence, these sites have become valuable sources of people's opinions and sentiments for businesses, researchers, and policymakers as more and more users post about services they use or express their political and religious views. According to a report by Pew Research Center, 69% of American adults used social networking sites in 2017, in contrast to only 5% using those sites in 2005 [Center, 2015]. Whereas these new communication channels, such as online social networks and news sharing sites offer myriad opportunities for knowledge sharing and opinion mobilization, they reveal an abundance of unfortunate intimidatory and hateful aggression towards individuals targeted [Willard, 2006] because of their identities or expressed opinions. These nasty and often coordinated victimizations of individuals have significant social costs ranging from social ostracism to opinion marginalization and suppression, and can cause severe individual health detriments such as anxiety, depression, and suicide ideation [Paul *et al.*, 2002]. A recent study [Duggan, 2014] found that 40% of adult Internet users have experienced online harassment with young women enduring particularly severe forms of it. 38% of women who had been harassed online reported the experience could be described as extremely or very upsetting. The U.S. Department of Justice statistics suggests that 850,000 American adults, mostly women are targets of cyberstalking each year, and 40% of women have experienced dating violence delivered electronically [Atlantic, 2014]. Victims of such online attacks are often minority groups or individuals voicing dissent, covering controversial topics essential for a democratic society, and raising awareness of uncomfortable information [Bernstein, 2014]. A study of female journalists examining off-line and online harassment found that two-thirds of the respondents reported acts of intimidation, threats, and abuse related to their work [Barton and Storm, 2014].

While several computational studies have developed automated mechanisms for detection of unwarranted victimization and harassing attacks on social network and microblogging platforms [Nobata *et al.*, 2016; Yin *et al.*, 2009;

Sood *et al.*, 2012] more comprehensive detection and intervention mechanisms that are grounded in well-founded, interdisciplinary theory of human aggressive and predatory behavior is needed. The goal of this paper twofold: (a) develop a nomenclature to characterize the different types of hateful and abusive rhetoric that is common online, (b) develop a range of Computational tools that can autonomously categorize individual communications and group predatory behaviors.

2 Background

Prior work on harassment detection spans several fields and several web platforms have been used as domains for detection which include: Twitter, Facebook, Instagram, Yahoo!, YouTubeDifferent platforms have unique purpose and content and may, therefore, display different subtypes of hate contents. For instance, one should expect quite different types of hate content on a platform catering to adolescents than on a web-platform used by a wider cross-section of the general public. Manual analysis of data and establishment of relationships between multiple features are often error-prone. Machine learning has been used to address this issue.

In [Yin *et al.*, 2009], a supervised classification technique is used along with local, sentimental and contextual features extracted from a post using Term Frequency-Inverse Document Frequency (TF-IDF). The classification technique is conjugated with n-grams and other features, such as incorporating abusiveness, to train a model for detecting harassment. A significant improvement over the general TF-IDF scheme is observedwhile adding the sentimental and contextual features. In [Sood *et al.*, 2012], Support vector machines (SVMs) were used to learn a model of profanity using the bag of words (BOW) approach to find the optimal features to be bigrams and stems by using a binary presence representation and a linear kernel. This approach surpasses the performance of all previously list-based profanity detection techniques. A linguistic and behavioral pattern based model [Mosquera *et al.*, 2014] was proposed to filter short texts, detect Spam and abusive users in the network. It used real-world SMS data set from a large telecommunications operator from the US and a social media corpus. It also addressed different ways to deal with short text message challenges such as tokenization and entity detection by using text normalization and substring clustering techniques. A comprehensive approach to detect hate speech was proposed in [Warner and Hirschberg, 2012] which presents a plan that targets specific group characteristics, including ethnic origin, religion, gender, and sexual orientation. The paragraph2vecapproach is used to classify anti-Semitic speech [Djuric *et al.*, 2015] on data collected over a 6-month period from Yahoo Finance website. In [Nobata *et al.*, 2016], Comprehensive lists of slurs, obtained from Hate speech and an array of features for abusive language detection (POS tags, the presence of blacklisted words, n-gram features including the token and character n-grams and length features) are used. Their scheme outperformed a deep learning approach by focusing on good annotation guidelines that help detect specific abusive language. In [Burnap and Williams, 2016], an exploratory single blended model of cyber hate that incorporates knowledge of features across multiple types was used. The proposed method improved classification for different types of cyber hate beyond the use of a BOW and known hateful terms. In [Waseem and Hovy, 2016] author analyzed the impact of various extra-linguistic features in conjunction with character n-grams for hate speech detection. It was observed that differences in the geographic and word-length distribution do not effect on performance and rarely improve over character level features but there is an exception to this with gender. A list of criteria based on critical race theory to identify racist and sexist slurs was presented.

In [Hosseinmardi *et al.*, 2015], a model to automatically detect incidents of cyberbullying over images in Instagram is presented. A collection of sample Instagram posts consisting of images and their associated comments have been used as a dataset. They demonstrate a Linear SVM classifier can significantly improve the accuracy of identifying cyberbullying by incorporating multi-modal features from the text, images, and metadata for the media session. In [Zhong *et al.*, 2016], author develop a method for detecting cyberbullying in commentaries following shared images on Instagram. For classification, they have used SVM with an RBF kernel and various feature sets. They have used Bag of Words, offensiveness score, LDA-generated topics from image captions, Clusters generated from outputs of a pre-trained Convolutional Neural Network over images is used for generating the feature set. Leveraging the features of the posted images, captions, and comments they achieved an accuracy of 93% to classify comments that contain bullying. They achieve an accuracy of 68.55% in the detection of images prone to cyberbullying.

A sentiment analysis based classifier [Gitari *et al.*, 2015] detects the presence of hate speech in web discourses such as web forums and blogs. It abstracted the hate speech into three main thematic areas of race, nationality, and religion. This model can identify subjectivity and rate the polarity of sentiment expressed in a given sentence.

Most of the research studies that we have come across have mostly focused on the binary classification of harassment as a communication is either harassment or not. Cyber Harassment is not a binary concept. Cyber harassment can be any form of interpersonal aggression sent using the web that may convey hostility, humiliation, insults, threats, unwanted sexual advances to a target. Cyber harassment behaviors can include offensive name-calling, attempts to embarrass, physical threats, stalking, gender harassment, unwanted sexual attention, sexual coercion, denigration, impersonation, flaming. As there is a gap between psychological concepts and computational models and concepts of what constitutes cyber harassment, training computers for comprehensive harassment detection is challenging. The goal of this paper is to remove this disconnect and leverage computational and Psychological approaches to identify different harassment categories and refine them using automated mechanisms, crowdsourced feedback, surveys and statistical techniques. The goals of this paper to build effective detection algorithms to identify harassment based on the identified categories.

3 Social Media Harassment Categories

A standard definitions of online harassment and aggression does not exist, but most definitions include the following key components: (1) unwanted behavior that occurs through electronically mediated communication (2) Behavior which violates the dignity of a person by creating a hostile, degarding, or offensive environment [Bossler *et al.*, 2012]. Behaviors can include offensive name calling, attempts to embarrass, physical threats, stalking [Duggan, 2014], gender harassment, unwanted sexual attention, sexual coercion, denigration (sending harmful or cruel statement about a person to other people online), impersonation (pretending to be another person in order to make that person look bad), flaming (sending angry, vulgar or rude messages about an individual through an online, public forum), and exclusion (the exclusion of an individual from an online group) [Staude-Müller *et al.*, 2012]. Definitions vary in terms of whether the behavior must occur multiple times, intentionally cause harm, and/or involve a perpetrator known to the victim [Duggan, 2014]. In [Newman *et al.*, 2016] have emphasized the need for clear definitions in research for the field to progress. Much of the current literature examining offline harassment require knowledge of a perpetrator's intentions, which while difficult to discern in offline environments are almost impossible to confirm in a computational behaviorally based model. For this reason, online harassment as currently understood by researchers is defined in terms of a victim or third party's understanding rather than a perpetrator's motives. With this is mind, researchers do believe there are three main reasons perpetrators may purposefully engage in harassing behavior. Purposefully harassing behavior includes (1) rude comments used as a form of self-expression(2) intimidation strategically designed to (a) interrupt communication on a topic or (b) retaliate for past reports or comments; and (3) acts with no strategic aims other than causing psychological or physical harm [Buckels *et al.*, 2014]. Further complicating an understanding of harassment is the variability in how harassment is perceived. Many definitions require the victim to view the behavior as offensive or threatening [Gidro *et al.*, 2016]. Understanding a victim's reaction is equally as difficult as discerning a perpetrator's motive when researchers are unable to directly communicate with the victim.

To understand different types of online harassment, a nomenclature was prepared containing different categories of cyber harassment that was paired with an associated vocabulary. The initial category list was generated from existing literature in psychology and listed as following: (i) other/general harassment (ii) cruel statements (iii) religious/racial/ethnic slurs (iv) harassment based on sexual orientation (v) sexual harassment (vi) threats of physical harm/violence (vii) multiple types (viii) non harassment.

4 Harassment Data set

We chose the Twitter platform for data scraping, using Twitter's streaming API, tweets matching to the keywords associated with the initially identified categories are collected and stored in underlying MySql database. As real-world data is often incomplete, inconsistent and likely to contain errors, collected data were cleaned and pre-processed to facilitate the research. After pre-processing the, 5230 out of 8000 collected tweets were usable. Then each tweet is coded by three labelers to make the data labeling more reliable. The result of the labeling is presented in the table 1. Almost 45% of usable data are categorized as non-harassing or do not fit into any given categories. We also observe that some of the categories have insufficient data for classification.

Table 1: Result of Data Labeling

Category	Data	Category	Data
General harassment	79	Cruel statement	1054
Religious/racial/ethnic	89	Sexual orientation	11
Sex/ gender	656	Threat	236
Multiple types	106	Non harassment	2382

5 Related Research Work

We build a labeled dataset that contains data from different categories. We use this labeled data and apply some of the state-of-the-art supervised learning algorithms such as Naive Bayes classifier, Etree and Support Vector Machine and proposed cluster-based categorization method to train and evaluate classification models for the target categories.

While constructing the Feature Vector for the state-of-the-art algorithms, we use two standard methods, which are (i) TF-IDF (ii) Count Vector and two methods built using the concept of Word2Vec[Kusner *et al.*, 2015] modeling: (i) Embedding and (ii) Embedding with TF-IDF.

TF-IDF [Salton and Buckley, 1988] or Term frequency-inverse document frequency is a widely-used approach in relevant document searching, text mining and information retrieval applications. TF-IDF weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is discounted by the frequency of the word in the corpus. The TF-IDF weight is typically composed of two terms: the first computes the normalized Term Frequency (TF), the number of times a word appears in a document, divided by the total number of words in that document; the second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears. *Term Frequency, TF:* TF measures how frequently a term occurs in a document. Since documents can vary widely in length, it is possible that a term would appear many more times in long documents than shorter ones. Hence, the term frequency is often normalized by the document length, i.e., the total number of terms in the document: $TF(t) = \frac{t_d}{T_d}$, where t_d is the number of times term t appears in a document and T_d is the total number of terms in the document. *Inverse Document Frequency, IDF,* is a measure of the importance of a term. While computing TF, all terms are considered equally important. However, it is known that certain common terms, such as "the", "is", "of", and "that", may appear frequently but have little importance. Hence, the weights of frequent

terms are discounted while that of the rare ones are increased: $IDF(t) = \frac{D}{D_t}$, where D is the total number of documents and D_t is the number of documents containing term t.

Unlike TF-IDF, no prior dictionary is needed in Countvector. It converts text documents to a matrix of the token(word) counts. This mechanism produces a sparse representation of the counts as there is no apriori dictionary, the number of features will be equal to the vocabulary size found by analyzing the data.

For the next two feature construction methods: (i) Embedding (ii) Embedding with TF-IDF, we use Word2Vec method. In Word2vec modeling, a two-layer neural net processes text and represents a set feature vector for each word in the text. The usefulness of Word2vec is to group the vectors of similar words together in vector space. Word2vec creates vectors that are distributed numerical representations of word features such as the context of individual words. Word2vec detects similarities between two words mathematically. Semantic vectors provided by Word2Vec preserve most of the relevant information about a text while having relatively low dimensionality which allows better machine learning.

Word2Vec model is built on standard Google NLP data set as well as the labeled data set. From the model, a dictionary is derived which maps each word to a 100-dimensional vector. These vectors are then used to build the features.

For the Embedding method, we build a feature vector by averaging the word vectors for all the words in a text.

In the Embedding with TF-IDF we apply the TF-IDF weighting scheme on top of Embedding to highlight the importance of the word. In this scheme, if a word was never seen, it would be at least as infrequent as any of the known words. So the default IDF is the max of known IDF's.

We have used Support Vector Machines (SVMs) as one of the classification algorithms to categorize the data. Classification of text data suffers from the curse of high dimensionality, fewer irrelevant features (features that can be discarded) and sparsity of document vectors. SVM can handle large feature spaces as it uses over-fitting protection which does not depend on the number of features.

One way to avoid these high dimensional input spaces is to assume that most of the features are irrelevant. Feature selection tries to determine these irrelevant features. Unfortunately, in text categorization, there are very few irrelevant features. SVM combines many features to learn a dense concept to overcome this challenge of text classification. For each document, the corresponding document vector contains only a few entries which are non-zero as the document contains a very small subset of the entire vocabulary. We regularized linear models with stochastic gradient descent (SGD) learning: the gradient of the loss is estimated for each sample and the model is updated along the way with a decreasing strength schedule. We use all the four feature construction methods and compare the results.

Naive Bayes classifier requires a small amount of training data to estimate the necessary parameters which makes it extremely fast compared to the others. Naive Bayes helps to alleviate problems stemming from the curse of dimensionality. Multinomial Naive Bayes implements the Naive Bayes algorithm for multinomially distributed data and is one of the two classic Naive Bayes variants used in text classification. The multinomial Naive Bayes classifier is suitable for classification with discrete features and hence is quite effective for text classification. The other classic Naive Bayes variant is Bernoulli Naive Bayes which is used for data that is distributed according to multivariate Bernoulli distributions; i.e., there may be multiple features but each one is assumed to be a binary-valued (Bernoulli, boolean) variable. Therefore, this class requires samples to be represented as binary-valued feature vectors; if handed any other kind of data, a Bernoulli Naive Bayes instance may binarize its input. We use all four Feature Vector construction mechanisms to classify the labeled dataset, for both the variants.

We have also used extra-trees regressors that fits a number of randomized decision trees on various sub-samples of the dataset as a meta estimator and use averaging to improve the predictive accuracy and control over-fitting. The algorithm uses the perturb-and-combine techniques designed for trees. This means a diverse set of classifiers is created by introducing randomness in the classifier construction. The prediction of the ensemble is given as the averaged prediction of the individual classifiers. In Extra-Tree Classifier, randomness goes one step further in the way splits are computed. Like random forests, a random subset of candidate features is used, but instead of looking for the most discriminative thresholds, thresholds are drawn at random for each candidate feature and the best of these randomly generated thresholds is picked as the splitting rule. This reduces the variance of the model at the expense of a proportional increase in the bias. The main parameters to adjust when using this classifier is the number of trees in the forest. A higher number of trees is better, although it will result in an increase in computational time. Additionally, beyond a critical number of trees, there will be no improvement in the performance. The size of the random subsets of features to consider when splitting a node is also a very important parameter. Here, a smaller size of the random subsets of features, results in a reduction of variance at the cost of an increase in bias.

6 Proposed Approach

In this section, we will introduce our new Word2Vec based text classification approaches. In particular, we propose two variations of cluster center based classification algorithm: one using a new vector weighing mechanism and the other adapting Word Movers Distance for cluster based classification.

To train a Word2Vec model we combine all text in the standard Google NLP data [Mikolov *et al.*, 2013] with the labeled harassment dataset that we have coded. The Word2Vec model generator builds a dictionary of words which maps each word to a n-dimensional vector (we chose $n = 100$) where words with similar meaning are mapped in the same neighborhood. We use this dictionary to form a semantic vector representation of each tweet.

6.1 Weighted vector clustering

In this variation of the algorithm we use a weighted Word2Vec representation. For a given tweet, we calculate the average of the weighted vectors for all the words in the tweet.

Algorithm 1 Cluster Based Classification

procedure TEXTREPRESENTATION(tweets,C)
  $WV = []$
  for do $t \in tweets$
    $c = C[t]$
    for do $w \in t$
      $WV[w] = Weight(w, t, c) * Word_{vect}(w)$
procedure TRAINING DATA(C, $train_{tweets}$)
  for do $n = 1\ to\ |C|$
    $Cluster[n] = []$
  for do $t \in train_{tweets}$
    $V = []$
    for do $w \in t$
      $V.append(WV[w])$
    $vector = Average(V)$
    $c = C[t]$
    $Cluster[c].append[vector]$
  $Center = []$
  for do $c \in Cluster$
    $Center[c] = Average(Cluster[c])$
procedure TESTING(C, $test_{tweets}$)
  for do $t \in test_{tweets}$
    for do $c \in Cluster$
      $V = []$
      for do $w \in t$
        $V[w] = Weight(w, t, c) * Word_{vect}(w)$
      $weighted_t = Average(V)$
      $dist[c] = distance(weighted_t, center[c])$
    $Label[t] = \arg\min_{c \in Cluster} dist[c]$

The distance between vectors corresponding to tweets that are semantically similar should be low as Word2Vec maps similar words to nearby points on a vector space

In the training phase, we know the category c of an input tweet t. For calculating the weight of a word, $w \in t$, we use the ratio of inverse document frequency of the word for the particular cluster that the word belongs to and the inverse document frequency of the word for overall data set:

$$Weight(w, t, c) = \frac{IDF_{class}(w, t, c)}{IDF(w, t)},$$

where $IDF_{class}(w, t, c)$ is the inverse document frequency of the word w for the particular cluster c

$$IDF_{class}(w, t, c) = \frac{n_c}{n_{c,t}},$$

where n_c is total numbers of tweets present in cluster c, and $n_{c,w}$ is number of tweets in the cluster c with term w in it. We also calculate the Inverse Document Frequency of t:

$$IDF(w, t) = \frac{T}{T_w},$$

where T is the total number of tweets and T_t is the number of tweets with the term w in it. In the training phase, we use the tweets for each class in training set to construct a cluster for that class. We calculate the weighted vector representation of the center of each cluster by averaging the weighted vector representation of all the tweets contained in that cluster.

In the testing phase, for each tweet in the test set, we calculate the distance between the center of each of the clusters and the weighted representation of the tweet. The weighted vectors of the words in a tweet are calculated based on the clusters. Then we calculate the distance between the center and weighted representation of the tweets, we choose the cluster which has minimum Euclidean distance.

One of the critical design choices of this algorithm is the choice of the distance measure between any two vectors. For the above algorithm, we have used Euclidean Distance to calculate the distance between the weighted vector representations. The algorithm is present in Algorithm 1.

6.2 Cluster Vector Algorithm

In this variant of the cluster based classification algorithm, we used vector representation without any weight. In particular, we used Word2Vec model to represent each word of the tweets. From this model, we get a dictionary which is mapping each word to a n-dimensional vector (we chose $n = 100$). We obtain a semantic vector representation for a given tweet by averaging the semantic vectors for all the words contained in the tweet.

In the training phase, we use the tweets in training set to first construct cluster for each of the categories. We calculate the vector representation of the center of each cluster by averaging the vector representations of all the tweets present in the cluster. We then find a given τ number (we chose $\tau = 3$) of closest tweets to the center of each cluster and these set of tweets are used as a representation of the cluster.

In the testing phase, for each tweet in the test set, we calculate the average distance between the tweets representing the center of any of the clusters and the tweet. We use Word Mover's Distance (WMD) [Kusner *et al.*, 2015] to calculate the distance between two tweets. WMD measure the dissimilarity between two text documents as the minimum amount of distance that the embedded words of one document need to move to reach the embedded words of another document. The cluster which has minimum average WMD distance with test tweet is considered to be its label. The algorithm is presented in Algorithm 2.

7 Results and Discussion

We employed the four feature construction mechanisms that we discussed earlier to train the classifiers. These mechanisms are: (i) Embedding: I, (ii) TFIDF embedding: II, (iii) Count-vector: III, and (iv) TF-IDF: IV. We used the labeled data set to train Multinomial and Bernoulli Naive Bayes classifier, Support Vector Machine, Etree classifier and both of our proposed algorithms. Finally these classifiers were evaluated for the target categories. To test algorithms, we compare their accuracy on the same dataset but in three different scenarios:

Binary Dataset, DSI: We constructed a binary data set from the labeled data set: all the tweets which fall into one of the preassigned harassment categories we consider to be of label '1' and all the tweets to be of label '0'.

Algorithm 2 Cluster Based Classification

procedure TRAINING DATA
  for do $n \in number\ of\ cluster$
    $Cluster[n] = []$
  for do $t \in train_{tweets}$
    $v = []$
    for do $w \in t$
      $V.append(Word_{vect}(w))$
    $vector = Average(V)$
    $c = C[t]$
    $Cluster[c].append[vector]$
  $Cen = []$
  for do $c \in Cluster$
    $Cen[c] = Average(Cluster[c])$
  for do $n \in number\ of\ cluster$
    $Center[n] =$ *the tweet closest to* $cen[n]$
procedure TESTING
  for do $t \in test_{tweets}$
    for do $n \in number\ of\ cluste$
      $dist[n] = WordMoversDistance(t, center[n])$
    $Label[t] = min(dist)$

Table 2: Classification accuracy of competing algorithms.

Algorithm	Data		
-	Binary	Multi-class	Multi-class
Multi NB I	0.7001	0.7005	0.66009
Multi NB II	0.7017	0.6905	0.61105
Multi NB III	0.70416	0.75	0.6692
Multi NB IV	0.7037	0.69	06175105
Bernoulli NB I	0.7005	0.07015	0.56056
Bernoulli NB I1	0.7175	0.70183	0.579802
Bernoulli NB III	0.70705	0.07185	0.5706
Bernoulli NB IV	0.70705	0.70705	0.5706
SVM I	0.6371	0.65699	0.5899
SVM II	0.68074	0.7302	0.6652
SVM III	0.6875	0.7572	0.654875
SVM IV	0.7185	**0.77709**	0.**6926818**
Etree I	0.71213	0.75318	0.6758788
Etree II	0.6946	0.710432	0.6547
Etree III	0.68882	0.74758	0.65089
Etree IV	0.7097	0.767938	0.6676
Algorithm 1	**0.722345**	0.656215	0.5481
Algorithm 2	0.6956	0.6408	0.53716

Multi-class - Non-Harassment dataset, DSII: In the second scenario we only considered the tweets which fall into one of the preassigned harassment categories. Which means the data set does not contain any tweets labeled as non-harassing.
Full Dataset, DSIII: In this case we considered all the labeled tweets.

To evaluate the classifiers, we used 10-fold cross-validation and calculated average accuracy. The results of the algorithms on the three data sets are presented in Table 2, with the best result of each dataset in bold.

For DSI, the cluster based algorithm with weighted vector achieved the maximum accuracy of 72.23%. While the cluster based algorithm with WMD achieved 69.56% accuracy. Both the variations of the Naive Bayes algorithm achieved around 70% for most of the feature construction mechanisms. Bernoulli Naive Bayes with TFIDF embedding achieved 71.71% of accuracy. Other than that, SVM with TFIDF achieved 71.85% and classifier with embedding achieved the accuracy of 71.21%. SVM with Embedding performed worst among all the algorithms and all the configurations.

For DSII, SVM with TF-IDF performed the best by achieving an accuracy of 77.71%. Etree classifier with TF-IDF achieved 76.79%. Bernoulli Naive Bayes performed better than the Multinomial Naive Bayes for three of the feature construction mechanisms. But the Multinomial Naive Bayes algorithm with Counter vector mechanism achieved accuracy at par with SVM with Counter vector mechanism.

For DSIII, SVM performed better compared to the other algorithms. The Performance of Etree classifier was at par with SVM.

In terms of relative performance, each classifier is able to perform marginally better than the others in different scenarios. Despite the several different types of features we tried, we were unable to achieve as high of an accuracy on the harassment data classification problem when compared to those reported for text classification. One limitation of the experiment was the size of the dataset. We believe that to be able to significantly increase the classification accuracy of categories of harassment, we need a much larger dataset. Another limitation has been that in the labeled data set, some of the harassment categories are quite sparse and disproportionately infrequent compared to the data which are labeled as non-harassing, which posed as a challenge to the learning mechanism. Thus, we need to expand the data collection process to generate a much larger data set that is well-balanced across the different categories of harassment.

8 Conclusions

While the preponderance of extensive and consistent use of cyber harassment and its negative impact on users and groups is well-recognized, the literature on formally characterizing the diverse types of online harassment lacks rigor. We have introduced a set of new categories of online harassment based on psychological constructs. We build a harassment dataset with communication retrieved from the TWitter platform and which are subsequently manually classified by multiple volunteer coders into one of the above identified categories. We then evaluate the effectiveness of a number of machine learning approaches, including some adapted from existing literature and a couple of new ones that we introduce, to detect the identified harassment types on this coded dataset. Results show different algorithms perform well on different versions of the dataset and also highlights the need for collecting a larger and more balanced dataset.

References

[Atlantic, 2014] Marlisse Silver Sweeney The Atlantic. What the Law Can (and Can't) Do About Online Harassment, (2014).

[Barton and Storm, 2014] Alana Barton and Hannah Storm. Violence and harassment against women in the news media: a global picture. *New York: Women's Media Foundation and the International News Safety Institute. Accessed November*, 6:2014, 2014.

[Bernstein, 2014] Anita Bernstein. Abuse and harassment diminish free speech. *Pace Law Review*, 35(1), 2014.

[Bossler *et al.*, 2012] Adam M Bossler, Thomas J Holt, and David C May. Predicting online harassment victimization among a juvenile population. *Youth & Society*, 44(4):500–523, 2012.

[Buckels *et al.*, 2014] Erin E Buckels, Paul D Trapnell, and Delroy L Paulhus. Trolls just want to have fun. *Personality and individual Differences*, 67:97–102, 2014.

[Burnap and Williams, 2016] Pete Burnap and Matthew L Williams. Us and them: identifying cyber hate on twitter across multiple protected characteristics. *EPJ Data Science*, 5(1):11, 2016.

[Center, 2015] Pew Research Center. Social Media Usage: 2005-2015, (2015).

[Djuric *et al.*, 2015] Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. Hate speech detection with comment embeddings. In *Proceedings of the 24th International Conference on World Wide Web*, pages 29–30. ACM, 2015.

[Duggan, 2014] Maeve Duggan. *Online harassment*. Pew Research Center, 2014.

[Gidro *et al.*, 2016] Romulus Gidro, Aurelia Gidro, et al. Aspects concerning sexual and moral harassment in the workplace. *Curentul Juridic, The Juridical Current, Le Courant Juridique*, 64:65–73, 2016.

[Gitari *et al.*, 2015] Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering*, 10(4):215–230, 2015.

[Hosseinmardi *et al.*, 2015] Homa Hosseinmardi, Sabrina Arredondo Mattson, Rahat Ibn Rafiq, Richard Han, Qin Lv, and Shivakant Mishra. Detection of cyberbullying incidents on the instagram social network. *arXiv preprint arXiv:1503.03909*, 2015.

[Kusner *et al.*, 2015] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966, 2015.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[Mosquera *et al.*, 2014] Alejandro Mosquera, Lamine Aouad, Slawomir Grzonkowski, and Dylan Morss. On detecting messaging abuse in short text messages using linguistic and behavioral patterns. *arXiv preprint arXiv:1408.3934*, 2014.

[Newman *et al.*, 2016] Elana Newman, Susan Drevo, Bradley Brummel, Gavin Rees, and Bruce Shapiro. Online abuse of women journalists: Towards an evidence-based approach to prevention and intervention. pages 46–52, 2016.

[Nobata *et al.*, 2016] Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*, pages 145–153. International World Wide Web Conferences Steering Committee, 2016.

[Paul *et al.*, 2002] Jay P Paul, Joseph Catania, Lance Pollack, Judith Moskowitz, Jesse Canchola, Thomas Mills, Diane Binson, and Ron Stall. Suicide attempts among gay and bisexual men: lifetime prevalence and antecedents. *American journal of public health*, 92(8):1338–1345, 2002.

[Salton and Buckley, 1988] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[Sood *et al.*, 2012] Sara Owsley Sood, Judd Antin, and Elizabeth F Churchill. Using crowdsourcing to improve profanity detection. In *AAAI Spring Symposium: Wisdom of the Crowd*, 2012.

[Staude-Müller *et al.*, 2012] Frithjof Staude-Müller, Britta Hansen, and Melanie Voss. How stressful is online victimization? effects of victim's personality and properties of the incident. *European Journal of Developmental Psychology*, 9(2):260–274, 2012.

[Warner and Hirschberg, 2012] William Warner and Julia Hirschberg. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26. Association for Computational Linguistics, 2012.

[Waseem and Hovy, 2016] Zeerak Waseem and Dirk Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of NAACL-HLT*, pages 88–93, 2016.

[Willard, 2006] Nancy Willard. Cyberbullying and cyberthreats. *Eugene, OR: Center for Safe and Responsible Internet Use*, 2006.

[Yin *et al.*, 2009] Dawei Yin, Zhenzhen Xue, Liangjie Hong, Brian D Davison, April Kontostathis, and Lynne Edwards. Detection of harassment on web 2.0. *Proceedings of the Content Analysis in the WEB*, 2:1–7, 2009.

[Zhong *et al.*, 2016] Haoti Zhong, Hao Li, Anna Squicciarini, Sarah Rajtmajer, Christopher Griffin, David Miller, and Cornelia Caragea. Content-driven detection of cyberbullying on the instagram social network. *IJCAI*, pages 3952–3958, 2016.