Collaborative Detection of Cyberbullying Behavior in Twitter Data

Amrita Mangaonkar, Allenoush Hayrapetian, Rajeev Raje
Department of Computer and Information Science
Indiana University-Purdue University Indianapolis
Indianapolis, USA
{apmangao@iupui.edu, ahayrape@umail.iu.edu, rraje@cs.iupui.edu}

***Abstract*— As the size of Twitter© data is increasing, so are undesirable behaviors of its users. One of such undesirable behavior is cyberbullying, which may even lead to catastrophic consequences. Hence, it is critical to efficiently detect cyberbullying behavior by analyzing tweets, if possible in real-time. Prevalent approaches to identify cyberbullying are mainly stand-alone and thus, are time-consuming. This research improves detection task using the principles of collaborative computing. Different collaborative paradigms are suggested and discussed in this paper. Preliminary results indicate an improvement in time and accuracy of the detection mechanism over the stand- alone paradigm.**

***Index Terms*—Cyberbullying, twitter, machine learning algorithms**

I. Introduction

Cyberbullying is one of the widely recognized problems increasing with the phenomenal growth of social media. Though the healthy social behavior is the solution to this problem, social media needs to consider integrating tools that can deal with this problem. The key to create these tools is an efficient detection of cyberbullying behavior on social networks. Social media data is extremely large, dynamic, interactive, and real-time. Therefore, tools that are used for detection of cyberbullying need to be faster as well. Current approaches, such as [2] [3], are sequential in nature and thus suffer from single-point failure and are slow in their performance. A distributed design for detecting cyberbullying is a promising approach to overcome these limitations of sequential approaches. This paper suggests cyberbullying detection of Twitter text data by analyzing the tweets in a collaborative and distributed manner. This collaborative paradigm evaluates and uses different machine learning techniques for classifying the tweets into either cyberbullying or non-cyberbullying behavior.

II. Motivation

The term "Cyberbullying" means, use of Information Technology to harm or harass other people in a deliberate, repeated, and hostile manner. It is different from traditional bullying as it can happen 24 hours of the day and seven days a week [1]. It happens in the form of mean text messages, spreading rumors, posting messages, and sharing embarrassing pictures and videos on social networking sites. Once such derogatory messages/pictures/videos are posted, it is very difficult to take these posts off the social media sites. Cyberbullying behavior is not only unacceptable but also can lead to catastrophic consequences. Hence, to have a safer and more constructive social environment, it is necessary to design a smart network or online patrol that will prohibit such behavior by monitoring and filtering the obscene, hateful, and improper content from the social media.

Unlike the prevalent approaches [2] [3], which are sequential in nature, a distributed paradigm is more suitable for detecting cyberbullying due to the following reasons: i) as the Twitter data is generated in distributed and asynchronous manner, it is better to detect the cyberbully behavior at different locations in a network, ii) a sequential detection technique will suffer from a single point failure, and iii) a distributed detection can reduce the analysis time by exploiting the inherent parallel nature associated with the independent generation of tweets.

III. Detection of Cyberbullying — Related Approaches

Various methods have been used for the detection of cyberbullying in a given textual content. These include Bag-of-Words (BoW), Lexical Syntactic Feature (LSF) [2], and different Machine Learning-based approaches [3]. Lexicon-based methods, such as the BoW or LSF, mainly rely on the presence of obscenities and profanities in the social media content. Although, textual cyberbullying may contain obscenities and profanities, all the obscene text on social media may not be cyberbullying – studies show that, rate of using offensive words is close to double on Twitter than in normal life [4]. Therefore, care must be taken in deciding whether a tweet constitutes cyberbullying or not, even if it contains obscenities. Over the years, different machine learning (ML) techniques such as Support Vector Machine, Naive Bayes, Maximum Entropy classification, have been used for document classification depending on sentiments [5]. The results of these experiments are encouraging and suggest that approach can be suitable for classifying a tweet either to

978-1-4799-8802-0/15/$31.00 ©2015 IEEE

be cyberbully or non-cyberbully behavior – and is used in the proposed research.

IV. Proposed Approach

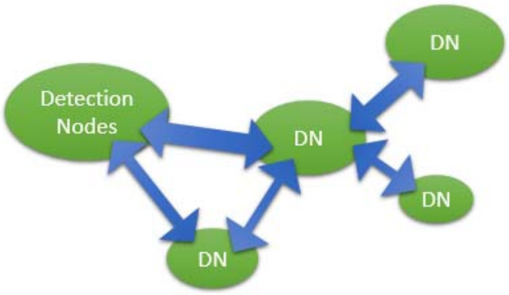

Fig. 1. Architecture of the Collaborative Cyberbullying Detection System

As indicated earlier, the proposed system employs a collaborative approach to classifying a tweet as “bullying” or “non-bullying”. The architecture of the proposed system is shown in Figure 1. Each entity, in Figure 1, is an autonomous detection node, and these nodes collaborate with each other when needed in an effort to classify a tweet as cyberbullying or not.

Prevalent approaches for detecting cyberbullying are sequential in nature and are also typically off-line. Applications such as Twitter© and Facebook® are inherently distributed as their data gets generated in a geographically dispersed and an asynchronous manner. As data is getting created in parallel, these applications get bombarded with incoming data from various sources in a very rapid manner. In such a situation, the sequential approach is certainly a drawback. Hence, the processing technique applied to this data must not only be quick and efficient, but it also needs to be able to adapt to a distributed environment. Thus, if the cyberbullying detection has to be made online, the detection process must begin before the data reaches the central server. This requires the detection mechanism to work in distributed mode, that is; there should be many nodes in the network that run detection algorithms in parallel. These nodes may collaborate with each other if needed, as shown in Figure 1. A detection node may collaborate with other nodes because i) the other detection node may be better equipped at classification, and/or ii) the other node may be able to provide a second opinion about a particular tweet.

In order to determine the actual algorithm employed in each detection node, various alternatives were experimented with. These alternatives include supervised machine learning algorithms such as Naive Bayes (NB), Logistic regression, and Support Vector Machine (SVM). These different algorithms were invoked on a publicly available tweeter dataset using the Weka (Waikato Environment for Knowledge Analysis) toolkit [6].

Detection nodes may run different algorithms; also they may use different training datasets. Datasets for training these algorithms were created by scrapping web pages and Twitter feeds that have reported cyberbullying instances. Non-bullying instances were collected directly from Twitter. These tweets were then manually labeled as “bullying” or “non-bullying” for the purposes of validation.

The first dataset created was balanced dataset and contained 170 bullying tweets and 170 non-bullying tweets. The second dataset was an imbalanced dataset, and it contained 177 bullying tweets and 1,163 non-bullying tweets. The reason for creating balanced and imbalanced dataset is to test the performance of aforementioned algorithms with different types of datasets. Imbalanced datasets closely resemble the real life tweet data that algorithms might need to train on. Thus, it is imperative that the learning capacity of ML algorithm is independent of skew in data.

V. Experiments with Machine Learning algorithms

Supervised machine learning techniques infer a classification function from labeled training data. Word vectors extracted from tweets play a key role for this experiment. Extraction of word vector from tweet is achieved by using a tokenizer. Then this word vector is used to relate to given output a value or a label. It is expected that an optimal scenario will allow an algorithm to determine the class labels correctly for each test dataset.

It is possible to fine tune the selection of this word vector that then forms an attribute set. This tuning is done by selecting parameters such as the *Minimum Term Frequency* and the *Tokenizer*. *Minimum Term Frequency* (MTF) allows filtering out words whose frequency in training dataset is below expected value. *Tokenizer* can be used to identify some phrases or sequence of words that always indicate bullying instances – e.g., “Go die.” The following parameter settings are used with each machine learning algorithm during the course of these experiments.

i) WordTokenizer with minimum frequency of word 1
ii) WordTokenizer with minimum frequency of word 2
iii) Bi-GramTokenizer with minimum frequency of word 1
iv) Bi-GramTokenizer with minimum frequency of word 2

The remainder of this section discusses the performance (as measured by the accuracy, precision and recall) of different ML algorithms (Naïve Bayes, Support Vector Machine and Logistic Regression) in a stand-alone mode. The results of the collaborative detection are described in Section VI. In both the cases, the outcome of the experiments is binary in nature – i.e., each tweet is classified either as “bullying” or “non-bullying”.

A. Experiments with balanced dataset

The first set of experiments consisted of the use of a balanced data set that contained 170 bullying and 170 non-bullying tweets. A balanced dataset has classification classes evenly represented, and there is no skew that will produce a bias for either of the class. Therefore, for the balanced nature dataset, it was expected that each ML algorithm will perform in a similar manner. Also, increasing the MTF should result in the increase of precision associated with the results as the word that frequently appears in bullying tweets get higher weights. However, the recall should reduce as the MTF increases because the words that might have been important for classification but appear less frequently in dataset than the set limit of the MTF, are filtered out. These results in reducing false positives. With bi-gram tokenizer, recall values are expected to be better, this is because bi-gram can quickly classify by identifying collocations. However, it might reduce precision as a result of the increase in false positives as the dataset is very small.

The outcomes of experiments, when the balanced dataset is used as the training set, are depicted below in Figures 2 and 3. F1 and F2 indicate frequency of word 1 and frequency of word 2 respectively. N2 indicates Bi-GramTokenizer is being used.

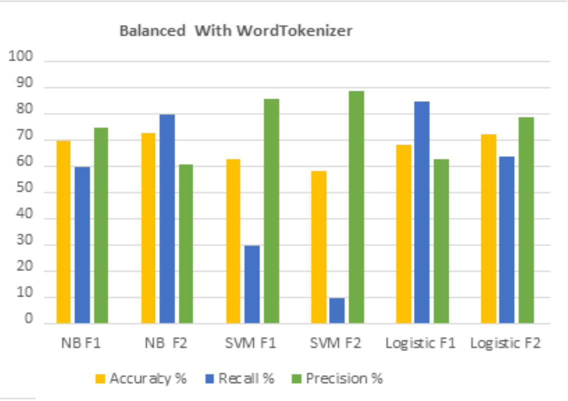

Fig. 2. ML algorithms performance comparison: balanced dataset

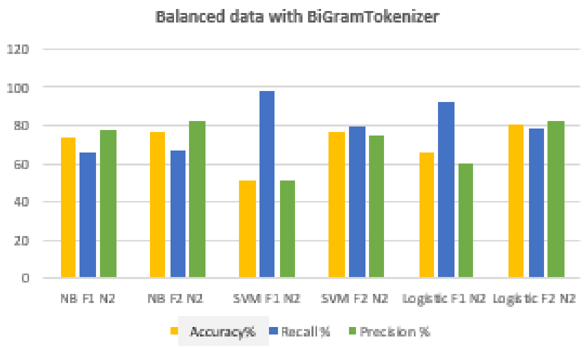

Fig. 3. ML algorithms performance comparison: balanced dataset

Analysis of outcome

As per the above mentioned expectations, all the ML methods do perform analogously. Also, **Bi-Gram** tokenizer does provide with better recall than wordtokenizer in almost all cases. Following is a detailed discussion of various methods.

i) **Naive Bayes:** This method displays more than 60% recall and precision for the balanced dataset. It also shows an improvement in accuracy, precision, and recall when bi-gram tokenizer is used, and word frequency is set to 2.

ii) **Support Vector Machine:** This algorithm is not able to catch enough positive cases without bi-gram. However, with the bi-gram tokenizer and word frequency of 2, the recall is improved by more than 70% that in turn increases the classification capabilities.

iii) **Logistic Regression:** This method produces coherent **recall than other two methods**, regardless of setting used for MTF or Tokenizer. The precision and recall are greater than 60% in all the cases for this dataset.

These results confirm the popular belief that the discriminative model (Logistic Regression) does perform a little better than the generative model (Naive Bayes). The SVM does produce better recall than the Logistic regression with Bi-gram and MTF 1 but with lesser precision. This is because, SVM finds the decision boundary by maximizing the margin between points closest to the classification. However, for given dataset, the boundary devised by SVM is trying to fit all the outliners of “bullying” class which decreases precision. Logistic Regression tries to maximize the likelihood that if a tweet belongs to “bullying” category depending on its word vector, then, unlike SVM, it does not need balancing between precision and recall.

B. Experiments with imbalanced dataset

The second set of experiment involved the use of the unbalanced data consisting of more non-bullying tweets than the bullying tweets. In case of unbalanced datasets, the SVM finds the decision boundary that maximizes the distance between the instances of two classes for a given application which fails to find this boundary [7] and thus, performs worse than the other two ML techniques – an observation that is confirmed by experimental results is indicated in Figures 4 and 5.

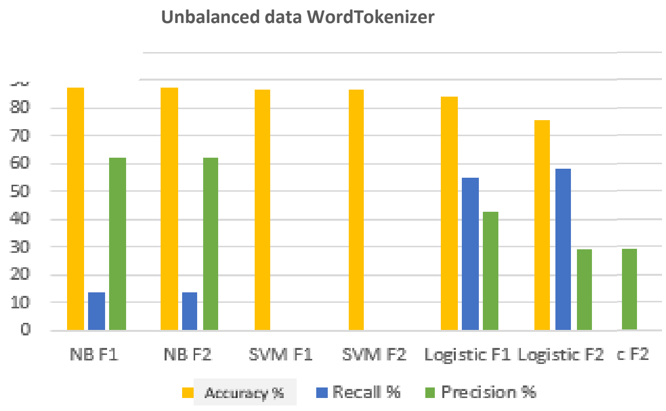

Fig. 4. ML algorithms performance comparison unbalanced dataset

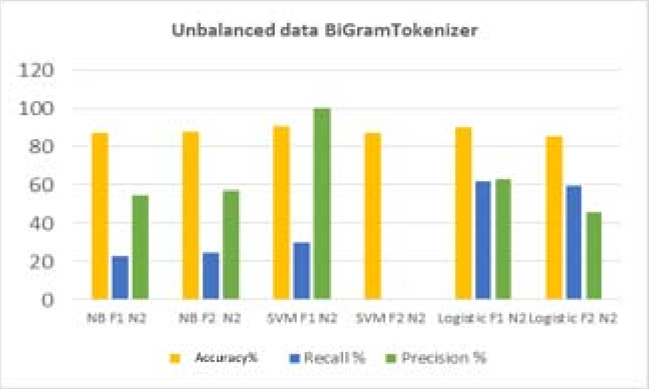

Fig. 5. ML algorithms performance comparison unbalanced dataset

Analysis of outcome

i) **Naive Bayes:** With an unbalanced dataset, the recall values have dropped; however, they are better than the ones obtained via the support vector machine.

ii) **Support Vector Machine:** It fails when classifying bullying tweets in unbalanced datasets [7].

iii) **Logistics Regression:** It captures a significant number of positive cases, and on an average more than 30% of the predictions are correct. It gives more than 50% precision and more 30% recall for both balanced and unbalanced datasets.

These experiments indicate that the Naive Bayes and the Logistic Regression algorithms are more suited for detecting cyberbullying behavior than the SVM. Naive Bayes being a generative machine learning algorithm learns the distribution of individual classes, while Logistic Regression being a discriminative algorithm learns the decision boundary. In accordance with the popular belief that discriminative algorithms work better than the generative ones [8], for this application, Logistic Regression seems to be working better.

VI. Distributed Collaborations

As indicated earlier, this research advocates the usage of a collaborative cyberbully detection paradigm over the sequential paradigm. Based on the outcome of the empirical evaluation, described in the previous section, it was decided that three different collaborative configurations can be used to carry out cyberbully detection:

i) **Heterogeneous Collaboration:** Each cyberbullying detection node has a different algorithm for detection.

ii) **Homogeneous Collaboration:** Each cyberbullying detection node has the same algorithm for detection. However, their knowledge bases (i.e., the training dataset) may or may not be the same. If a method is typically performing better than others, then it is better to use the homogeneous collaboration than the heterogeneous collaboration.

iii) **Selective Collaboration**: A node always sends a message to an expert node. The expert node is decided by history.

An immediate consequence of the collaborative paradigm is the need to merge the classification results obtained from different detection nodes. To combine results of above collaborations following methods are used:

i) **AND Parallelism**: In this case, all detection nodes and/or algorithms work on a single tweet in parallel. This logic is similar to an exhaustive collaboration.

ii) **OR Parallelism**: All detection nodes and/or algorithms work on a single tweet in parallel. Once any of these nodes classifies the tweet as bullying, all other detection activities are stopped.

VII. Case Study of Homogeneous Collaboration with Different Knowledge Base at each node

The first case study used the homogeneous configuration with each node employing the logistic regression technique. The logistic regression algorithm was chosen for each node as it had better performance, as indicated in Section V, amongst machine learning techniques. However, each detection node was trained on different data. This test intents to observe if such collaboration can improve the accuracy of detection of cyberbullying and also reduce the time required for this detection.

A. Test setup

The aforementioned unbalanced dataset was divided into five parts. Each part was skewed, and there was no overlapping data existed. Four detection nodes (or servers) were used for this experiment. Of these five datasets, one was being chosen every time as a test dataset. Each classifying server was then trained to one of the remaining datasets. Word Tokenizer was used, and minimum frequency of the word was set to 2.

To combine the results of collaboration, following three different techniques were used:

i) **AND parallelism:** Each tweet was classified by all four severs. If more than half of the servers (in the case of four servers, more than two servers) classify a tweet as “bullying, then it was considered as a “bullying” tweet.

ii) **OR parallelism:** Each tweet was sent for classification to all four servers. If any one of them classifies it as “bullying” then the others were stopped and the tweet was considered as a “bullying” tweet.

iii) **Random 2 OR Parallelism:** Each tweet was sent to two random servers for classification. The classification was based on OR parallelism between these two servers.

These collaborations were then compared with the setup where all the data was used to train a single server, which was then used for the classification of a test set. The comparison was based on Precision, Recall, Accuracy, and Time as the performance parameters.

B. *Outcomes of experiment*

This section describes two different experiment setups and their outcomes.

Test setup 1

Table 1 provides details of the datasets used in the first set of collaborative experiments. The test set selected for this setup contains 207 non-bullying tweets and 31 bullying tweets. All other datasets are used as training sets.

TABLE 1: DATASETS CONFIGURATION 1

Set number	Negative Cases	Positive Cases	Skew
Train Data 1	202	42	17%
Test Data	207	31	13%
Train Data 3	189	23	10%
Train Data 4	336	49	12%
Train Data 2	83	12	12%

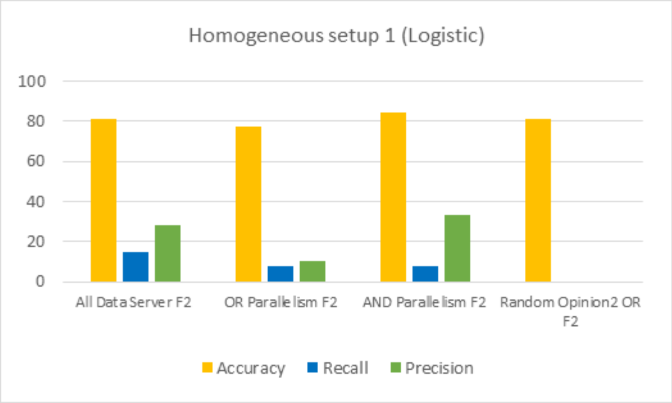

Fig. 6. Classification results for homogeneous collaboration with Logistic Regression algorithm at detection nodes. (Setup-1)

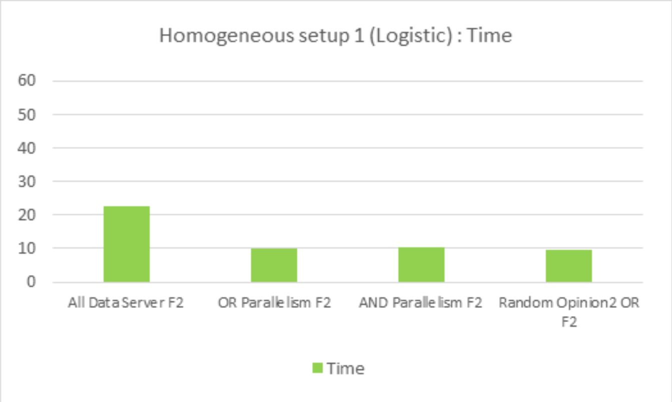

Fig. 7. Time taken for classification with homogeneous collaboration with Logistic Regression algorithm at detection nodes (Setup-1)

Test setup 2

Table 2 provides details of the datasets used in the second set of collaborative experiments. The test set selected for this setup contains 202 non-bullying tweets and 42 bullying tweets. This way it can be analyzed that, whether experiments will produce similar results for datasets with different data and skew.

TABLE 2: DATASETS CONFIGURATION 2

Set number	Negative Cases	Positive Cases	Skew
Test Data	202	42	17%
Train Data 1	207	31	13%
Train Data 3	189	23	10%
Train Data 4	336	49	12%
Train Data 2	83	12	12%

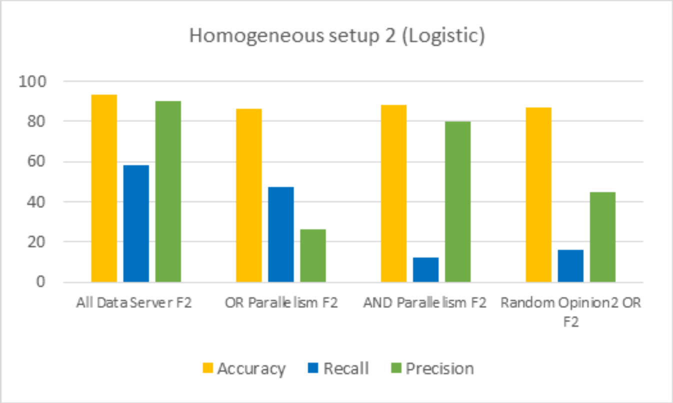

Fig. 8. Classification results for homogeneous collaboration with Logistic Regression algorithm at detection nodes (Setup-2)

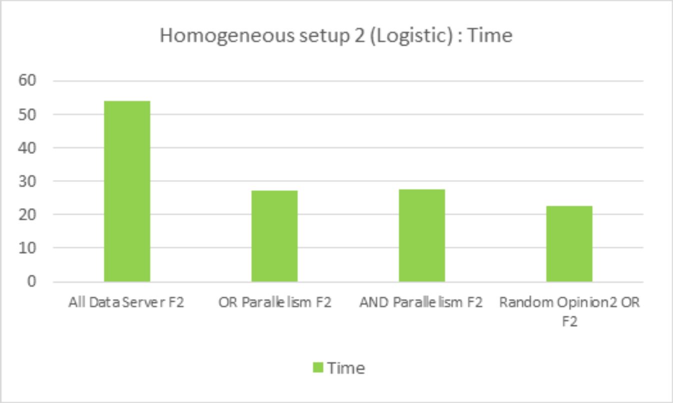

Fig. 9. Time taken for classification with homogeneous collaboration with Logistic Regression algorithm at detection nodes (Setup-2)

C. *Observation and results*

As expected, the “All Data Server” (i.e., a single server which contains all the data) took longer to classify the same amount of tweets as compared to any collaborative techniques as shown in Figures 7 and 9. The reason behind this behavior is that the collaborating servers were working in parallel, and each had a smaller feature set to parse through.

From Figures 6 and 8, it can be seen that, “All Data Server” yielded better recall values, because it is just one node that has all the data, as opposed to having part of it on multiple nodes with less knowledge about it. However, it did not always guarantee better accuracy and precision.

The AND parallelism produced better precision in 4 out of 5 cases. The precision is better when false positive cases are less in number. In the case of AND parallelism, if one of the servers incorrectly classifies the data, then it is covered by three other servers. The final result of this process yields a correct classification, thus, making false positives less in number while increasing the overall precision in most of the cases. On the other hand, when only one server does correct classification, its results are wiped-out by other servers, which leads to more false negatives, and thus, the recall is seen less in AND parallelism. Same thing happened in the case where AND parallelism could not produce better precision.

In collaboration techniques, OR parallelism had the best recall results while the AND parallelism had the best accuracy. The recall is better when false negatives are less in number. In the case of OR parallelism, if any one server classifies a tweet as bullying then it is considered as bullying. Hence, any of the servers classifying the tweet as bullying, suffices, thus decreasing false negative cases and increasing recall. However, when any of them classifies a tweet incorrectly as bullying, it is considered as bullying, which increases the number of false positives and thus reducing overall precision.

Figures 7 and 9 illustrate that the classification time taken by the collaboration techniques was almost half the time taken by the “All Data Server” technique. Referring to Figures 6 and 8 again, it can be observed that the precision obtained with the collaborative technique is better than the server trained with the entire knowledgebase in 4 out of 5 cases. Accuracy in the collaborative technique worked better in 3 out of 5 cases. Accuracy increases and, as a result, false positives and false negatives are reduced in number. This happens as collaborations tend to get balanced results.

VIII. Conclusion And Future Work

In conclusion, 7 out of 15 cases using collaboration techniques worked better than their sequential counterpart. These results were achieved without much tuning of algorithms. Therefore, if the results obtained via the collaborative technique are comparable with the sequential approach, then it is worth using the collaborative techniques because it is more efficient time-wise.

Future work includes an analysis of other collaboration techniques. It is also necessary to check the above hypothesis (i.e., collaboration is preferable over the sequential paradigm) for a larger dataset. An analysis of lexicon-based and machine learning techniques in a combined setting should also be considered. Also, in the above experiments, the history of the relationship between two Twitter accounts has not been considered. Consideration of such a history will be another direction for future work.

References

[1] "What is Cyberbullying". http://www.stopbullying.gov/.

[2] Chen, Y., Zhou, Y., Zhu, S., and Xu, H. “Detecting offensive language in s ocial media to protect adolescent online safety” Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Conference on Social Computing (SocialCom), pp. 71-80.

[3] K. Dinakar, B. Jones, C. Havasi, H. Lieberman and R. Picard “Common sense reasoning for detection, prevention, and mitigation of cyberbullying” ACM Trans. Interact. Intell. Syst., vol. 2, no. 3, pp. 29-31, 2012.

[4] Steinmetz, K. "#Cursing Study: 10 Lessons About How We Use Swear Words on Twitter". time.com 19 Feb 2014.

[5] Pang, B., Lee, L., & Vaithyanathan, S. (2002, July). “Thumbs up?: sentiment classification using machine learning techniques.” In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (pp. 79-86). Association for Computational Linguistics.

[6] University of Waikato. “Weka 3: Data Mining Software in Java.” http://www.cs.waikato.ac.nz/ml/weka/.

[7] Akbani, R., Kwek, S., & Japkowicz, N. (2004). “Applying support vector machines to imbalanced datasets.” In Machine Learning: ECML 2004 (pp. 39-50). Springer Berlin Heidelberg.

[8] Ng, A. Y., & Jordan, M. I. (2002). “On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes.” Neural Information Processing Systems 14.