Semi-supervised Learning for Cyberbullying Detection in Social Networks

Vinita Nahar[1,⋆], Sanad Al-Maskari[1], Xue Li[1], and Chaoyi Pang[2]

[1] School of Information Technology and Electrical Engineering,
The University of Queensland, Australia
{v.nahar,s.almaskari}@uq.edu.au,xueli@itee.uq.edu.au,
[2] The Australian E-Health Research Center, CSIRO, Australia
Chaoyi.Pang@csiro.au

Abstract. Current approaches on cyberbullying detection are mostly static: they are unable to handle noisy, imbalanced or streaming data efficiently. Existing studies on cyberbullying detection are mainly supervised learning approaches, assuming data is sufficiently pre-labelled. However this is impractical in the real-world situation where only a small number of labels are available in streaming data. In this paper, we propose a semi-supervised leaning approach that will augment training data samples and apply a fuzzy SVM algorithm. The augmented training technique automatically extracts and enlarges training set from the unlabelled streaming text, while learning is conducted by utilising a very small training set provided as an initial input. The experimental results indicate that the proposed augmented approach outperformed all other methods, and is suitable in the real-world situations, where sufficiently labelled instances are not available for training. For the proposed fuzzy SVM approach we handle complex and multidimensional data generated by streaming text, where the importance of features are discriminated for the decision function. The evaluation conducted on different experimental scenarios indicates the superiority of the proposed fuzzy SVM against all other methods.

Keywords: Cyberbullying Detection, Text-Stream Classification, Semi-supervised learning, Social Networks.

1 Introduction

Current studies on cyberbullying detection are mainly focused on: i) Supervised learning approaches, which rely on a human-intensive labelling process of data. ii) Feature space is uniformly applied to a learner. Whereas, streaming text generated by Social Networks (SNs) is highly uncertain, noisy, and imbalanced. In such a changing environment, different training data samples may have varying levels of importance. Therefore, with the rapid growth of user-generated content in SNs, existing supervised approaches become unaffordable and impractical for

⋆ v.nahar@uq.edu.au

H. Wang and M.A. Sharaf (Eds.): ADC 2014, LNCS 8506, pp. 160–171, 2014.
© Springer International Publishing Switzerland 2014

automatic detection of cyberbullying instances. In this paper, we focus on the detection of cyberbullying in streaming text generated by SNs. For such detection the following challenges are identified:

(i) ***Insufficient training instances:*** Streaming text arriving from SNs, is either seldom labelled or not labelled at all. Moreover, it is impractical to ask users to label the messages into cyberbullying and non-cyberbullying categories. Therefore, as an alternative to manual labelling of the entire streaming text, only a small set of labelled instances are available for training.
(ii) ***Uncertain and imbalance feature distribution:*** All the input features is not evenly important for the learners' decision function. For example, the baseline swear-keyword based feature 'hell' is often used in normal communication. Using such words may not increase the discriminating effectiveness of the learner. To mimic the real-world situation, highly unstable and imbalanced data are fed to the system.

To address above challenges, we emphasis on cyberbullying detection under semi-supervised learning approaches. For this work, we assume that only a small set of labelled instances are available for initial system training. We consider two methods: i) based on augmented training by using ensemble classifiers with a confidence voting function. A confidence voting function is defined based on the parameter Γ to extract and enlarge training set from the unlabelled streaming text automatically. ii) using a fuzzy SVM algorithm to cater for the uncertain or irrelevant nature of the dynamic and multidimensional input feature space. In FSVM there are N possible free parameters (N), where N is the total number of training points. A degree of importance s_i is given to each data point providing a greater flexibility and generalisation to the model. For each training pair (x_i, y_i) a membership value s_i is given; the pairs with high s_i values will have a greater influence in the decision surface compared to the one with lower s_i values.

2 Related Work

Recently, Xu *et al.* explored *regret* behaviours in bullying messages assuming that people who posted bullying tweets may later want to delete those posts [1]. They reported cross validation accuracy upto 60.7%. Dadvar *et al.* used content-based, cyberbullying, and user-based feature sets [2]. The best recall obtained (55.0% recall, 77.0% precison, and 64.0% F_1 *measure*) with user-based and pronoun-profanity window feature sets. Dinakar *et al.* deconstructed cyberbullying detection into sensitive-topic detection, which is likely to result in bullying discussions, including sexuality, race, intelligence, and profanity[3]. Using SVM, the accuracy archived is 79% under the topic sexuality. Nahar *et al.* utilised probabilistic features and user ranking, and achieved 99% accuracy [4]. Yin *et al.* utilised various features including content, sentiment, and contextual features, showing 59.5% recall, 35.2% precision, and 44.4% F_1 *measure* [5].

However, these methods are conducted under supervised learning by directly applying the whole input feature space to a learner. These techniques are unable to handle the imbalanced and noisy data, where some features are either

irrelevant or less important for the decision function. In this paper, we introduce semi-supervised learning for cyberbullying detection in streaming text.

3 Methodology

3.1 Feature Space Modeling

To understand the semantic structure users could have in mind while posting a comment, various features can be helpful. An enriched feature set can be generated from the given posts. These features are commonly known as linguistic features and are used mostly in nature language processing applications. These features are defined as follow: (i) Keywords based features, which involve binary representation of the keywords to see if the keywords are presented or not; (ii) To capture the influence of malevolence within messages, we also used the normalised value of the keywords. It is the number of swearwords in posts, divided by the total number of the words in messages; (iii) Presence of pronouns such as, 'you' and 'he' which makes the message more personal. For instance, if the keyword appears near 'you', it will likely indicate that the message is more targeted towards that person. Yei *et al*. used pronouns as sentiment features [5] for harassment detection, where normalised values of second and third person pronouns are used; (iv) To capture a degree of users' emotions, emotions are included for the feature space design. Normalised values of happy and angry emotions are computed separately for each comment; (v) Mostly, people on the Internet use capital letters to indicate that they are yelling or shouting. The normalised value of capital letters within messages is used to capture the loudness; (vi) Some other meta data of messages, such as special characters, are used in their normalised form; and (vii) Users' age and gender are also used as features because the selection of words, usage, and language vary between people of different age groups and gender.

In addition to above features, we also extracted location information. However in the dataset, most users come from different places of the USA, which would not carry ethnic or cultural differences since they are all from the same country. Therefore, location-based features are not considered for such datasets. However, it will be interesting to include location information to capture a certain degree of the ethnic or cultural differences and the language styles used.

3.2 Cyberbullying Detection

Following the traditional practices for performing text-stream classification [6][7], we assume that the unlabelled data streams U_n of varying length are arriving on the system in sessions. A very small set of positive instances and some negative instances are available for initial training ($P_n \cup N_n$). To ensure that negative instances do not undermine the decision function for positive instances, random under-sampling of negative instances is adopted at the initial training phase. Every session is trained on two base classifiers. The model automatically extracts

strong positive and some negative instances (T'_n) to enlarge the training set (T_n). The extraction and enlargement step is computed by combining the decision of both the classifiers using voting. To combine the decisions of multiple classifiers by voting, there are various ways such as linear combination, majority voting etc. that can be used. Out of those linear combination is the simplest way to combine multiple classifiers. According to the linear combination of the learners:

$$y_i = \sum_j \alpha_j d_{ji} \quad where, \quad \alpha_j \geq 0, \sum_j \alpha_j = 1 \tag{1}$$

Another possibility to find w_j is by assessing classifiers accuracy from separate feature set and use the information to compute the weight. We define *confidence function* Φ to predict the class label of the input instance given by Equation 2:

$$\Phi = \prod_j \alpha_j d_{ji} \quad where, \quad \alpha_j \geq 0, \sum_j \alpha_j = 1 \tag{2}$$

In the $confidence\ function$ (Φ) α_j is the weight of the vote of the base classifier j for class C_i and d_{ji} is the vote of base classifier j for class C_i. Φ is defined based on the product of the probability distribution for the class C_i for the given test instance. As we are interested in cyberbullying posts, we select the predicted positive class based on the probability distribution of the base classifier j.

$$y_i = \begin{cases} 1 \ \ if\ \Phi \quad \geq \quad \Gamma \\ -1\ if\ \Phi < (1-\Gamma) \end{cases} \tag{3}$$

In U_n, cyberbullying instances are rare compared to normal or non-cyberbullying instances therefore, adding all the identified negative instances may results in overfitting. Thus, for each added positive instance $y_i = 1$, only two negative instances $y_i = -1$ will be augmented in the enlarged training set, T'_n.

We employ two base classifiers g_1 and g_2, which are well-known text classifiers. g_1 is a Naive Bayes multinomial text classifier, and the second base classifier g_2 is a Stochastic Gradient Descent classifier. Both classifiers are extended from WEKA[1], as they are available in WEKA under function-based algorithms.

The second base classifier g_2 is built using the Stochastic (or "on-line") Gradient Descent for text (SGD text) classifier [8] as an implementation in WEKA. The SGD text algorithm is designed for the text data. It generates feature space using an STWV filter to transform text strings into term-weight vectors based on Vector Space Model [9]. We use default values of the SGD text i.e. support vector machines as the loss function, learning rate of 0.01, a regularisation constant of 0.01, 500 iterations without pruning the dictionary, 3 as a minimum word frequency, default normalisation and no transformation to lower case of the input instances. The tokenisation of text strings is performed with the tokenise module specified by a parameter. It is an iterative method, which builds the learning model iteratively i.e. training is conducted in successive sessions until the algorithm converges using a loss function.

[1] `http://www.cs.waikato.ac.nz/ml/weka/`

Algorithm 1 : Model building using Augmented training set

Input:
P_n: Small set of positive instances for initial training;
N_n: Small set of negative instances for initial training;
U_n: Set of unlabelled examples of the incoming session;
Γ : *parameter*;
Output:
T_n: Enlarged strong training set
1. $T_n \leftarrow P_n \cup N_n$;
2. //Extraction step:
3. Train C_1 by T_n;
4. Train C_2 by T_n;
5. $T'_n \leftarrow \{C_1, C_2\}$, Γ, using Equation 3;
6. //Enlargement step:
7. $T_n \leftarrow T_n \cup T'_n$;
8. *return* T_n;

By Using Fuzzy Approach: Given that the streaming text generated from the SNs is highly uncertain, complex, and unbalanced, we incorporate a membership generation method of the robust FSVM model. For the given array of text features for each user's comment, the method should have strong discriminatory power capable of ranking the input feature space. There are various methods used to generate membership values [10], [11], [12]. We use a Kernel-based Fuzzy C-Means (K-FCM) clustering algorithm to generate memberships values for our fuzzy classifier model owing to its ability to handle noise and outliers.

Clustering Process: The first step in K-FSVM is to cluster the incoming pre-processed text-stream data sets. In a complex and dynamic environment such as SNs, a range of features can be generated from single user post. Each feature will have a different degree of information and relevance to a specific concept therefore, calculating the total relevance of each instance from all features is highly important for the learning model. To achieve this goal we employ a fuzzy clustering approach, which enables us to evaluate all features and calculate their degree of relevance to a specific group. Clustering is used to find high intra-cluster and low inter-cluster similarities. The idea is to find natural groupings among similar objects.

Kernel-based FCM was introduced to overcome noise and outliers sensitivity found in FCM [13], [14] by transforming input space X to a high or infinite dimension feature F space ($\phi : X \rightarrow F$). For non-linearly separable problems, the input data can be projected to a high-dimensional feature space using a kernel. According to Cover's theorem, projecting input data into a high dimensional feature space is assumed to convert non-linearly separable problems into linearly separable in the feature space. This idea has been utilised in unsupervised

learning and by many algorithms including RBF Networks, SVM and other non-linear discriminating techniques. In a Kernel FCM the input space is projected to higher dimensional feature space using RBF, polynomial kernel or any other kernel type.

A Kernel-based Fuzzy C-Means clustering (KFCM) algorithm has been proposed by Zhang *et al.* [10][11]. KFCM partitions a given data set $X = \{x_i, ..., x_n\} \in R^p$ into $Cfuzzy$ fuzzy subsets by minimising the following objective function:

$$J_m(U,V) = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^m ||\phi(X_K) - \phi(V_i)||^2 \tag{4}$$

Subject to:

$$\sum_{k}^{n} u_{ik} > 0, \forall i \in 1, ...c \tag{5}$$

$$\sum_{i}^{c} u_{ik} > 1, \forall k \in 1, ...n \tag{6}$$

where, c is the number of clusters determined initially ($1 < c < n$). According to the condition in Equation 4 no cluster is empty; n is the number of data points; u_{ik} is the membership of X_k in class i satisfying $\sum_i^c u_{ik} = 1$ for all k and $u_{ik} \in [0, 1]$; m is the quantity controlling cluster fuzziness ($m > 1$); V is a set of control cluster centres or prototypes ($V_i \in R^p$); ϕ is an implicit nonlinear transformation function. The Euclidean distance between points and centres in the feature space F can be computed as:

$$||\phi(X_K) - \phi(V_i)||^2 = k(X_K, X_K) + k(V_i, V_i) - 2k(X_K, V_i) \tag{7}$$

where, $K(X,Y) = \phi(X)^T\phi(y)$ is an inner product of the kernel function X denotes the data space, and $\phi(x) \in F$, where $x \in X$, F is the transformed feature space and $K(x,x) = 1$. In our case, a Guassian Kernel was adopted, where $K(x,y) = exp(-d(x,y)^2/^2)$. Hence for $K(x,x) = 1$, the Gaussian Kernel leads to $d\phi^2(x,y) = K(x,x) + K(y,y) - 2K(x,y) = 2(1 - K(x,y))$. Thus the objective functions in Equation 4 becomes:

$$J_m(U,V) = 2\sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^m (1 = k(X_K, V_i)) \tag{8}$$

where,

$$k(X_K, V_i) = exp(-||X_K - V_i||^2/\sigma^2) \tag{9}$$

The optimisation problem is solved by minimising $J_m(U,V)$ under the constraints of u_{ik}.

$$u_{ik} = \frac{(1/(1 - k(X_K, V_i)))^{1/(m-1)})}{\sum j = 1^c (1/(1 - k(X_K, V_i)))^{1/(m-1)}}, \quad \forall i \in 1...c \; and \; \forall k \in 1...n \tag{10}$$

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^m K(X_K, V_i) X_K}{\sum k = 1^n u_{ik}^m K(X_K, V_i)} \tag{11}$$

One of the critical steps in Fuzzy based SVM is the membership generator method. From the previous step the membership matrix is generated and used by a fuzzy classifier decision function. A good membership matrix should degrade the effect of outlier and noise, and improve overall classification results. The following algorithms are used to generate the membership matrix for the fuzzy classifier:

Algorithm 2: Kernel Fuzzy C-Mean Clustering Algorithm

Input:
b_n: Streaming text;
m: Set Fuzzification parameter;
c: Number of clusters;
ε_1:set termination parameter;
Output:
Membership matrix
1. Select the kernel function K and its parameters;
2. Select cluster centres v_i
3. Update membership matrix u_{ik} using Equation 10;
4. Compute all new clusters or prototype V_i using Equation 11;
5. Repeat step 3-4 and check the termination function E^t;
6. $E^t = max|v_new - V_old|$, if $E_t \leq \varepsilon$, **stop**;
7. *return* membership matrix;

Fuzzy Classifier: The enriched feature space generated in cyberbullying context will have some training points with a varying level of importance. Consequently, the training points with higher impact should be classified correctly and the noisy points or meaningless ones will not be considered and discarded. This indicates that one point can belong 85% to one class and the remaining 15% can be meaningless or 10% can belong to one class and 90% can be meaningless. In contrast to SVM, fuzzy SVM allows each data point X_i to be assigned a membership value, U_i where $0 < U_i \leq 1$. The membership U_i is used to determine the importance or relativity of each data point X_i to one class and the value $(1 - U_i)$ can be used to determine the degree of meaningless.

In this section, a fuzzy classifier is used to handle unbalanced and unstable text streams generated from social networks. The dataset is fed into the KFCM model to extract membership values, and then a one-versus-one (OVO) fuzzy SVM model is constructed. Each membership value is used by the FSVM for classification. It is expected that noise will be assigned a low membership degree and each membership value will be used by the FSVM model resulting in better generalisation and accuracy. The following steps are performed to execute the K-FSVM model:

Algorithm 3 : Fuzzy SVM Classification algorithm

Input:
b_n: Streaming text;
s_n: Membership matrix generated by Algorithm 2;
ε_1: Set termination parameter;
Output:
Final prediction matrix;

1. Use OVO strategy to create multiple classifiers initialize all parameters including kernel function, cluster number, termination parameter ε and membership m;
2. Apply the memberships s_n to FSVM model. Here, each data point will have one membership value.The new training set will have x, y, u, where u is the membership value for x data point.
3. Predict all class labels using voting by classifying which classes are receiving most voting;
4. *return* final classifier;

4 Experiments Setting

4.1 Dataset

In the experiments, we utilised data provided by Fundacion Barcelona Media[2] for the workshop on content analysis from the Web 2.0. The given data was collected from the three different SNs including *Myspace*, *Kongregate*, and *Slashdot*. Characteristics of data from these three sites are different from each other. Our task was to extract cyberbullying instances from the streaming text of any type. The raw data was available in XML file format of different structures. A data parser was developed to extract the content, time, and user information. During the feature space modelling, extensive pre-processing was conducted in order to remove insignificant features.

4.2 Evaluation

The classification of cyberbullying messages is a critical issue because of different impacts made by the false positives and the false negatives. On one hand, to identify non-cyberbullying instances as cyberbullying itself is a sensitive issue (the false positive). On the other hand, the system should not miss out the cyberbullying posts (the false negatives). Though the false positive and the false negative instances are both critical, the ideal scenario is to achieve a high recall. That means cyberbullying-like posts should not be overlooked by the system

[2] `http://caw2.barcelonamedia.org/` Retrieved 10 November 2010.

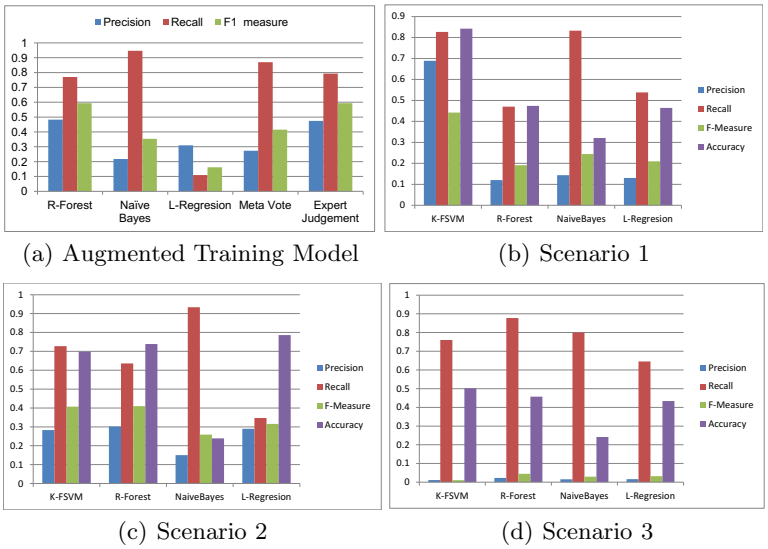
- a strict approach. Nevertheless, we present a performance metric including precision, recall and F_1 *measure* for evaluation. Table 1 shows the distribution of positive instances, r in the training and testing dataset experimental setting for various scenarios used.

4.3 Results and Discussions

Experiment 1: We employ the session scenario by sorting messages using time information and generated N streaming sessions of varying length. For experiments, we select parameters, $N = 75$ and $\Gamma = 0.95$. The final training set constructed by the augmented training method is used as a training set. To evaluate this model on a test set (manually validated test set), we employ Random Forest, Naïve Bayes, Logistic Regression, and Meta classifiers, and results are shown in Figure 1(a). Expert judgement is also presented to compare the other classifiers. The majority of feature selection methods work better if the frequency of the positive-like features is high. From Figure 1(a), we observe that the model is able to capture likely positive words including words that appear in the keyword list. While detecting cyberbullying in social networks, recall is a critical evaluation matrix as it is very important to reduce false negatives. The system should not be able to misclassify positive cases as negative - that is, cyberbullying-like cases are not ignored. As shown in Figure 1(a), Random Forest performed similarly to that of the expert judgement, with 48% precision, 77% recall, and 59% F_1 *measure*. In this experiment, the false positive is higher than the false negative. If we try to reduce the false positive, then the false negative increases. This is because discrimination of the positive features and the negative features is very vague. In the training data, we observe that many likely cyberbullying words are quite frequent in both cyberbullying and non-cyberbullying categories. Ignoring those words on one hand reduces the false positives, while on the other hand it increases the false negatives. Our objective is to reduce the false negatives; therefore our system tolerates the false positives but maintains low false negatives. Nevertheless, in this experiment the objective was to achieve high Recall, which is achieved up to 79.3%.

Table 1. Positive Instance Distribution, r

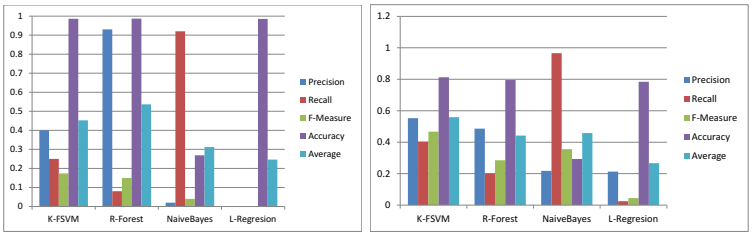
Experiment	r	
	Training	**Testing**
Scenario 1	34.0%	13.2%
Scenario 2	42.1%	14.2%
Scenario 3	34.0%	1.5%
	10-fold cross validation	
Scenario 4	1.5%	
Scenario 5	22.0%	



(a) Augmented Training Model (b) Scenario 1

(c) Scenario 2 (d) Scenario 3

Fig. 1. Results of Experiments 1 and 2



(a) Scenario 4 (b) Scenario 5

Fig. 2. Results of Experiment 3

Experiment 2: In this experimental set up, experiments are conducted in three different scenarios, when $r = 13.2\%$, 14.2%, and 1.5% respectively. The results are shown in Figures 1(b), 1(c), and 1(d).

In Scenario 1 ($r = 13.2\%$), K-FSVM outperformed all other methods. Recall is 82% while maintaining 6% precision. Nevertheless, Naïve Bayes achieved the highest recall (83%), which is almost the same as K-FSVM, whereas, Random Forest and Logistic Regression both performed similarly well.

In Scenario 2 ($r = 14.2\%$), the overall performance of K-FSVM and Random Forest are almost same, with the F_1 *measure* at 41%. While K-FSVM achieved higher recall, Random Forest achieved higher precision. Naïve Bayes achieved a 93% recall, which is one of the major requirements for a cyberbullying detection system. As discussed earlier, we are interested in high recall. That means the actual number of genuine cyberbullying cases identified by the system out of

all the genuine cyberbullying cases should be as high as possible. At this stage, the system may have an increased number of false alarms (the false positive), but it will not overlook cyberbullying instances. Logistic regression performed reasonably well in this scenario when compared to all others scenarios.
In Scenario 3, when $r = 1.5\%$ only, the system has achieved a very high recall as shown in Figure 1(d). This indicates that the system did not let the cyberbullying-like posts go unnoticed, although precision is poor. Though the false positive instances are high, the false negatives have been reduced significantly. Indeed, it is worth having a high number of the false positives identified by the system rather than ignoring genuine or cyberbullying-like posts.

Experiment 3: In scenarios 4 (r=1.5%) and 5 ($r = 20\%$), experiments are conducted using 10-fold cross-validation. In this evaluation setup, the complete dataset is partitioned ten times into 10 samples. In every round, randomly, nine sections are selected for training and the remaining section is used for testing. However, in such cases it is possible that the training phase may not be able to catch positive instances. In fact, this likelihood increases when the positive instances are rare, which is true in our case. For this reason we also decide to compare overall performance, which is an average of precision, recall, F_1 *measure* and accuracy. From Figure 2(a), very interesting results are observed. Overall K-FSVM achieved the best results in both experiments. Moreover, in Scenario 4, when the positive to negative ratio is 1.5%, Random Forest maintains a very high precision at 93%, whereas, Naïve Bayes achieved the highest recall 92%. Such observation opens a future direction to combine both of these classifiers to improve the systems performance significantly. In scenario 5, K-FSVM outperformed all other methods in terms of precision (55%) and F_1 *measure* (47%), whereas, Naïve Bayes achieved 97% recall and Logistic regression achieved poor results.

5 Conclusions

This paper proposed a semi-supervised approach for detection of cyberbullying in SNs. Our contributions can be summarised as: (i) We devised a new framework for automatic detection of cyberbullying for the streaming data with insufficient labels. The framework extracts reliable positive and negative instances by augmented training methods based on the confidence voting function. (ii) The enriched feature sets were generated based on user context, linguistic knowledge, and baseline keywords were also incorporated during feature space design in the proposed method. (iii) We also proposed a fuzzy SVM algorithm for the effective cyberbullying detection. The proposed method effectively tackles the dynamic and complex nature of the streaming data. (iv) The experiments conducted under the different scenarios demonstrate that the proposed technique outperformed the traditional methods use for cyberbullying detection.

References

1. Xu, J.M., Burchfiel, B., Zhu, X., Bellmore, A.: An examination of regret in bullying tweets. In: The 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 697–702 (2013)
2. Dadvar, M., Trieschnigg, D., Ordelman, R., de Jong, F.: Improving cyberbullying detection with user context. In: Serdyukov, P., Braslavski, P., Kuznetsov, S.O., Kamps, J., Rüger, S., Agichtein, E., Segalovich, I., Yilmaz, E. (eds.) ECIR 2013. LNCS, vol. 7814, pp. 693–696. Springer, Heidelberg (2013)
3. Dinakar, K., Reichart, R., Lieberman, H.: Modeling the detection of textual cyberbullying. In: AAAI Conference on Weblogs and Social Media, pp. 11–17 (2011)
4. Nahar, V., Unankard, S., Li, X., Pang, C.: Sentiment analysis for effective detection of cyber bullying. In: Sheng, Q.Z., Wang, G., Jensen, C.S., Xu, G. (eds.) APWeb 2012. LNCS, vol. 7235, pp. 767–774. Springer, Heidelberg (2012)
5. Yin, D., Xue, Z., Hong, L., Davisoni, B.D., Kontostathis, A., Edwards, L.: Detection of harassment on web 2.0. In: Content Analysis in the Web 2.0 Workshop at WWW (2009)
6. Zhang, Y., Li, X., Orlowska, M.: One-class classification of text streams with concept drift. In: ICDMW, pp. 116–125 (2008)
7. Nahar, V., Li, X., Pang, C., Zhang, Y.: Cyberbullying detection based on text-stream classification. In: AusDM (2013) (in press)
8. Zhang, T.: Solving large scale linear prediction problems using stochastic gradient descent algorithms. In: ICML, pp. 919–926. ACM (2004)
9. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)
10. Zhang, D.Q., Chen, S., Pan, Z.S., Tan, K.R.: Kernel-based fuzzy clustering incorporating spatial constraints for image segmentation. 4, 2189–2192 (2003)
11. Zhang, D.Q., Chen, S.C.: A novel kernelized fuzzy c-means algorithm with application in medical image segmentation. Artificial Intelligence in Medicine 32, 37–50 (2004)
12. Krishnapuram, R., Keller, J.M.: A possibilistic approach to clustering. IEEE Transaction on Fuzzy Systems 1, 98–110 (1993)
13. Wong, C.C., Chen, C.C., Yeh, S.L.: K-means-based fuzzy classifier design. 1, 48–52 (2000)
14. Gröll, L., Jäkel, J.: A new convergence proof of fuzzy c-means. 13, 717–720 (2007)