Using KNN and SVM Based One-Class Classifier for Detecting Online Radicalization on Twitter

Swati Agarwal and Ashish Sureka

Indraprastha Institute of Information Technology, Delhi (IIIT-D)
New Delhi, India
{swatia,ashish}@iiitd.ac.in

Abstract. Twitter is the largest and most popular micro-blogging website on Internet. Due to low publication barrier, anonymity and wide penetration, Twitter has become an easy target or platform for extremists to disseminate their ideologies and opinions by posting hate and extremism promoting tweets. Millions of tweets are posted on Twitter everyday and it is practically impossible for Twitter moderators or an intelligence and security analyst to manually identify such tweets, users and communities. However, automatic classification of tweets into predefined categories is a non-trivial problem problem due to short text of the tweet (the maximum length of a tweet can be 140 characters) and noisy content (incorrect grammar, spelling mistakes, presence of standard and non-standard abbreviations and slang). We frame the problem of hate and extremism promoting tweet detection as a one-class or unary-class categorization problem by learning a statistical model from a training set containing only the objects of one class . We propose several linguistic features such as presence of war, religious, negative emotions and offensive terms to discriminate hate and extremism promoting tweets from other tweets. We employ a single-class SVM and KNN algorithm for one-class classification task. We conduct a case-study on Jihad, perform a characterization study of the tweets and measure the precision and recall of the machine-learning based classifier. Experimental results on large and real-world dataset demonstrate that the proposed approach is effective with F-score of 0.60 and 0.83 for the KNN and SVM classifier respectively.

Keywords: Mining User Generated Content, One-Class Classifier, Online Radicalization, Short-Text Classification, Social media analytics, Twitter.

1 Research Motivation and Aim

Twitter[1] is a popular social networking website and the largest micro-blogging platform on Internet. Twitter allows users to share ideas and information instantly by posting short messages of 140 characters called as Tweets. Research

[1] https://twitter.com/

R. Natarajan et al. (Eds.): ICDCIT 2015, LNCS 8956, pp. 431–442, 2015.

shows that Twitter has become a platform for online radicalization and posting hate and extremism promoting content due to low publication barrier, lack of stringent moderation, anonymity and wide penetration [2][4][9].

Automatic identification of hate and extremism promoting tweets is useful to intelligence and security informatics agents as well as Twitter moderators. Manual identification of such tweets and filtering information from raw data is practically impossible due to the large volumes of tweets (500 million) posted every day. Tweets consists of short text (maximum of 140 characters) and noise (incorrect grammar, spelling mistakes, slang and abbreviations) as a result of which automatic classification of tweets is a technically challenging problem. The motivation of the work presented in this paper is to investigate solutions to address the problems encountered by intelligence and security informatics agents and Twitter moderators for countering online radicalization on the largest micro-blogging platform on Internet. The research aim of the work presented in this paper is the following:

1. To investigate techniques to automatically identify hate and extremism promoting tweets. To identify linguistic and stylistic features and characteristics of hate and extremism promoting tweets.
2. To conduct empirical analysis on a large real-world dataset and demonstrate the effectiveness of the proposed Machine Learning based text classification approach. To examine the relative influence of each proposed feature for the task of identifying hate and extremism promoting tweets. To compare and contrast the performance of various Machine Learning algorithms (KNN and LibSVM) for the purpose of recognizing hate and extremism promoting tweets.

2 Related Work and Research Contributions

We conduct a literature survey in the area of online radicalization detection on Web 2.0 (refer to Table 1) and textual classification of microblogs (refer to Table 2). Online radicalization, hate and extremism has been studied on multiple topics and domains: terrorism, anti-black communities, nationalism, politics, jihad and anti-Islam. Table 1 and 2 also mentions the experimental dataset size used in each study. We characterize papers based on the linguistic features used for the classification task and highlight the tweet classification goal: humor [10], irony [10], sarcasm [7], spam [8], vulgarity [12] and sentiments. There are many categories in tweet classification but due to page limitation we discuss a few of them here. Table 3(a) and 3(b) shows the dimensions for reviewing these categories and features respectively. We conclude from the related work that there is a research gap in the area of hate and extremism promoting tweet classification (intersection of online radicalization on Web 2.0 and short text or micro-blog classification). In context to existing work, the study presented in this paper makes the following unique contributions extending our previous work ([1]):

Table 1. Summary of Literature Survey of 9 Papers on Detecting Various Forms of Radicalization on Twitter

Ref	Year	Study	Objective	Dataset	
				Tweets	Nodes
[2]	2013	Nationalism	Identification of most influential, active and engaged hate promoting accounts on Twitter.	342K	3.5K
[4]	2013	Anti-Black	Classification of racist and non-racist conflicts in tweets by applying statistical measures.	24.5K	-
[9]	2013	Nationalism	Identification and analysis of extreme right communities on various social networking websites.	-	1697
[11]	2013	Terrorism	Content analysis of tweets in order to identify hidden groups related to a specific topic.	-	-

Table 2. Summary of Literature Survey of 11 Papers on Classifying Tweets

Ref	Year	Research Study							Objective	Features											Data
		C1	C2	C3	C4	C5	C6	C7		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	
[7]	2013		✓						A linguistic analysis based approach to filter **sarcastic** tweets.						✓		✓	✓	✓		3.38M
[10]	2012	✓		✓	✓				Identifying **irony** and **humorous** message on Twitter.			✓	✓		✓			✓		✓	50K
[12]	2012	✓				✓			Identification of inappropriate and **vulgar** language in tweets.								✓			✓	696M
[8]	2012							✓	Language model to filter **spam** tweets in most trending topics.	✓	✓			✓	✓	✓	✓			✓	20M
[13]	2012						✓		Discovering valuable tweets with of **interest** to its audience.	✓			✓		✓	✓		✓			64M

Table 3. List of Tweet Classification Goals and Linguistic Features

(a) **Categories**

Symbol	Summary
C1	Sentiment Classification
C2	Sarcasm Classification
C3	Irony Classification
C4	Humor Classification
C5	Offensive Tweets
C6	Interestingness
C7	Spam Detection
C8	News & Public Opinion
C9	Software Related
C10	Political Preference
C11	Writing Style

(b) **Features**

Symbol	Summary
F1	Direct Message
F2	Shortened URLs
F3	Emoticons
F4	Punctuations
F5	Topics
F6	Hashtags
F7	Mentioned Entities
F8	N-grams
F9	+ve and -ve Comments
F10	Emphasis (CAPS)
F11	Terms

1. A one-class classifier for identifying hate and extremism promoting tweets. While there has been work done in the area of humor, sarcasm, irony, sentiment, vulgar and spam tweets, to the best of our knowledge, our study is the first work on hate and extremism promoting tweet identification using a one-class classifier framework.
2. An empirical analysis on real-world Twitter dataset investigating the influence of various linguistic features (discriminatory features) for the task of recognizing hate and extremism promoting tweets. We conduct a series of

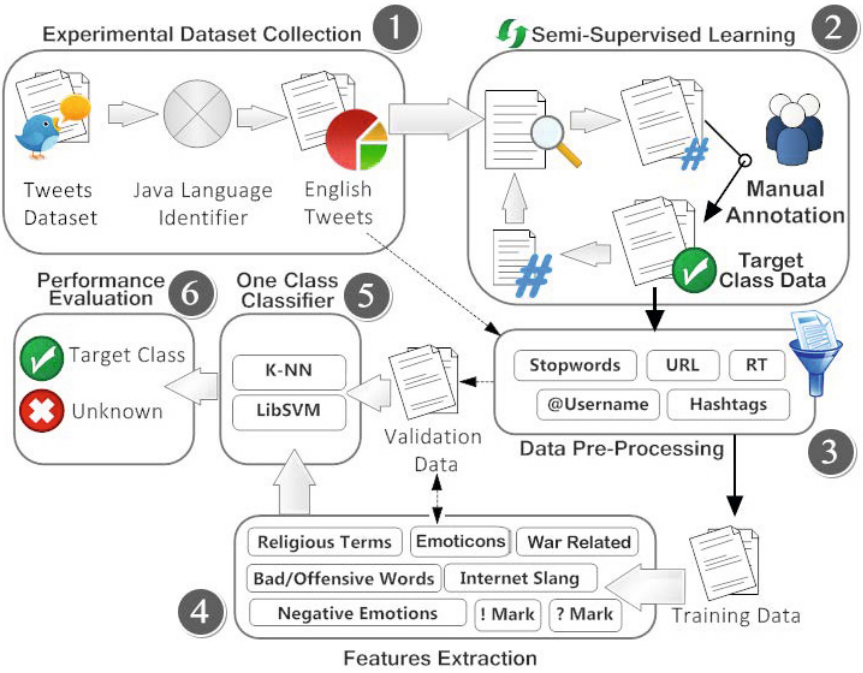


Fig. 1. A General Research Framework For Our Proposed Solution Approach

experiments to train a one-class SVM and KNN classifier and test its effectiveness for the given classification task.

3 Research Framework

Figure 1 illustrates the proposed solution approach. The proposed method is a multi-step process primarily consists of six phases: experimental dataset collection, training dataset creation, data pre-processing, feature extraction, one-class classification and performance evaluation. The six phases are labeled in the solution framework. In phase 1, we download two publicly available datasets [5] [6] (refer to Section 5.1 on experimental dataset) and combine them to form a single experimental dataset (a larger and diverse dataset to generalize our results). The dataset consists of tweets belonging to multiple languages. We use a language detection library[2] to filter English and non-English tweets. We conduct experiments only on English language tweets and discard non-English tweets. We notice that 85% of tweets are in English.

We require training dataset to create a statistical model for one-class classification task of identifying hate and extremism promoting tweets. We use

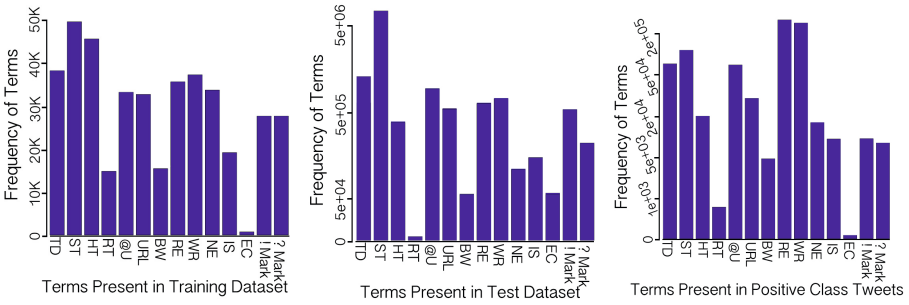
[2] `https://code.google.com/p/language-detection/`

Table 4. A Sample of Hate Promoting Tweets Leading to More Hashtags

Seed Hashtag	Tweet	Extended Hashtags
#Terrorism	Secret #recruitment British students #Muslim #extremists ? #islamophobia **#terrorism**	#islamophobia, #extremists
#Islamophobia	#NoJihad #Racism lowest form stupidity ! **#Islamophobia** height common sense ! #Quran	#NoJihad, #Racism
#Extremist	Engaging #AfPak Information War: Countering **#extremist** #propaganda with #mobile #technology	#propaganda
#Islam	**#Islam** evil according #GeertWilders one few islamophobic people Netherlands yet everywhere	#GeertWilders
#Terrorism	New: Al Qaeda Bomb Maker Video **#terrorism** #bomb #video #alqaeda #alquida	#bomb, #alqaeda, #alquida

Table 5. A Sample of Keywords Present in Hate Promoting Tweets

Hashtags	#islamophobia, #stealthjihad, #myjihad, #extremists, #NoJihad, #terrorism, #dreamact, #terrorist, #nativist, #GeertWilders, #alqaeda, #assassination
Religious	hijab, hizb, demon, jihad, god, maulana, kabba, azan, burka, prophet, koum, apostate, sikh, muhajir, immigrant, hijr, amen, hinduism, devil, atheist
War Related	LOC, Bomb, Blast, Attack, Holy war, Warfare, Tribute, Soldier, Jawan, Refugee, Enemies, Fighting, Patriot, Assassination, Expose, Army, Zindabad
-ve Emotions	endangered, enslaved, entangled, evaded, evasive, evicted, excessive, excluded, exhausted, exploited, exposed, fail, fake, hatred, regret, disgust, flaw
Emoticons	:), :-), :D, :-D, =], :], ;), =P, :P, :-P, :*, :(, :-(, =(, :-S, :S, :O, :-O, :\, :-\, \-o, :-}X, :-(, =), :-E, :-F, :-C, 3:*>, :-(, :(, :-d, :->, :-@,)8-), 3:), O:), :'(
Internet Slangs	LOL, haha, ROFL, WT*, WTH, IMHO, OSM, AKA, BRB, 404, CC, TC, TT, Cya, Gr8, FAQ, FYI, Hw, L8r, N/A, W/O, B/W, BTW, NP, OMG, PLZ
Bad Words	ahole, ass, ba****d, bit*h, crap, f**k, gay, damned, hells, jackoff, sh**, pe***, sexy, sl*t, XXX, b17ch, s.o.b., wh**e, screw, bulls**t, d-bag, jerk-off



(a) Training Dataset (b) Testing Dataset (c) Positive Class Tweets in Testing Dataset

Fig. 2. Frequency Distribution of Various Terms Present in the Training and Testing Dataset. TD= Tweet Dataset, ST=Stopwords, HT= Hashtags, RT= Retweets, @U= @username Mentioned, URL= Hyperlinks, BW= Bad Words, RE= Religious Terms, WR= War Related Terms, NE= Negative Emotions, IS= Internet Slangs, EC=Emoticons.

a semi-supervised learning[3] approach to create our training dataset. Annotating Tweets is a time-consuming and tedious task (practically challenging to annotate a large dataset). Hence, we use semi-supervised learning making use of a small amount of labeled data and a large amount of unlabeled data. Hashtags are strong indicators of the topic of the tweet. We create a list of seed hashtags such as #Terrorism, #Islamophobia and #Extremist and identify tweets containing these hashtags. We manually analyze tweets containing such hashtags and identify hate and extremism promoting tweets. We extend the list of hashtags by extracting new hashtags (not already in the list) present in the positive class tweets. Table 4 illustrates a sample of some seed hashtags and their respective tweets leading us to new hashtags. We then identify tweets containing the new hashtags and manually analyze the tweets to identify hate promoting tweets. As a result of this, we extend the list of hashtags and our training dataset of size S. We repeat this process several times to collect training dataset. This dataset and list of hashtags is publicly available at `https://sites.google.com/a/iiitd.ac.in/agrswati/datasets`.

We make our experimental dataset publicly available so that our experiments can be replicated and used for benchmark purposes by other researchers. We perform a random sampling on English tweets and use a sample as our testing (or validation) dataset. We remove the term 'RT' (Re-Tweet), @username (username of the direct mention of a user in the tweet), URL (short URL) and hashtags. After removing these terms our problem becomes more challenging due to short text classification. In phase 4, we perform characterization and identification of various discriminatory features and compute the frequency (TF) of various terms. For example, religious, offensive, slang, negative emotions, punctuations and war related terms. Table 5 shows a sample of these terms present in hate and extremism promoting tweets. Figure 2(a) and 2(b) shows the frequency of these terms present in the training and testing dataset. While Figure 2(c) shows statistics of only positive class tweets present in testing dataset. Figure 2 also illustrates the frequency of terms that have been preprocessed in phase 3. All statistics are computed in logarithmic scale. These graphs shows that the frequency of religious and war related terms is very high in hate promoting tweets.We convert our datasets (training and testing) into a matrix of feature space; where each entity represents a TF of respective column feature in a given tweet.

In phase 5, we implement two independent one-class classifiers (KNN and LibSVM) to classify a tweet as hate promoting or unknown. We use LibSVM as it is a popular open-source machine-learning library implementing the SMO (Sequential Minimal Optimization) for SVMs supporting classification and regression. Algorithm 1 & 2 describes the procedure of KNN and LibSVM classifiers respectively. In last phase, we evaluate the performance of the two classifiers using standard confusion matrix.

[3] Semi-Supervised Learning: learning the classifier from a combination of both labeled and unlabeled data.

Algorithm 1: One Class K-NN Algorithm

Data: Training Dataset D_{tr}, Test Dataset D_{te}, Neighbors K, Threshold th
Result: List of class labels for test dataset C_{te}
Algorithm $OneClassKNN(D_{tr}, D_{te}, th, K)$

1 **for** ***each*** *instance* $I \in D_{te}$ **do**
2 $N1 \leftarrow NearestNeighbor(I, D_{tr})$
3 $D_1 \leftarrow Euclidean\_Distance(N, I)$
4 **if** $(K == 1)$ **then**
5 $N2 \leftarrow NearestNeighbor(N1, D_{tr})$
6 $D_2 \leftarrow Euclidean\_Distance(N1, N2)$
else
7 $ND_1 \leftarrow Euclidean\_Distance(D_{tr}), N1)$
8 $D_2 \leftarrow Average(ND_1, ND_2, ..., ND_K)$
end
9 **if** $(D_1/D_2 > th)$ **then**
10 $C_{te}.addClass(Unknown)$
else
11 $C_{te}.addClass(TargetClass)$
end
end
12 return C_{te}

4 Solution Implementation

4.1 K-Nearest Neighbor Classifier

The proposed method (Algorithm 1) follows the standard one class KNN algorithm in order to classify a tweet as hate promoting or unknown. Inputs to this algorithm are pre-processed training dataset D_{tr}, testing dataset D_{te}, number of nearest neighbors K and a threshold measure th for accepting outliers. Each tweet in testing dataset is an arbitrary instance I that is represented by a feature vector $(f_1(I), f_2(I), ..., f_m(I))$ where $f_i(I)$ is an instance value for given feature and m is the number of discriminatory features. In steps 2 and 3, we compute euclidean distance[4] between an instance I of testing data and all instances of training datasets.

$$D = \sqrt[2]{\sum_{i=1}^{n}(f_i(I) \sim f_i(J))^2}, \text{ where } J \in D_{tr} \quad (1)$$

We create a distance matrix of size $n * 1$ for every instance $I \in D_{te}$, where n is the size of training dataset. Equation 1 shows the formula for computing euclidean distance between two instances. Based upon this distance matrix we find a nearest neighbor N_1 of I in training data. In steps 4 to 7, we find K nearest neighbors of N_1 in training dataset D_{tr}. Due to the large size of testing dataset we use $K = 100$. In step 8, we take an average of all K distances and name it as

[4] http://en.wikipedia.org/wiki/Euclidean_distance

D_2. Steps 9 to 11 perform unary classification. If the ratio of distances D_1 and D_2 comes out to be lower than threshold measure th, then instance I belongs to the target class otherwise it is classified as unknown. We compute an extent of similarity (euclidean distance) between all instances of training dataset D_{tr}. As a result of this, we get a distance matrix of size $n * 1$. We take a harmonic mean of these distances and come up with the threshold value th.

4.2 Support Vector Machine Algorithm

One class SVM is a supervised learning technique that performs distribution estimation of given training dataset. We develop an algorithm that classifies most positive class tweets from outliers. In our research, we use LibSVM java library 3.18[5] for Weka 3.7.10[6], originally proposed by Chang et. al. [3]. LibSVM is a wrapper class that allows one class SVM classifier supported by LibSVM tool. In one class LibSVM, all SVM formulations are supported as quadratic minimization problem. Equation 2 shows the formulation of unconstrained dual form of standard SVM classifier, subject to a Lagrange multiplier α that varies between 0 & a constant value C. Q is an n*n matrix where n is the size of training vectors and e is a vector of all ones represented as [1,1,....,1]. To constraint in minimization we optimize margin hyperplane as $y^T\alpha = 0$. In one class LibSVM (Equation 3), we solve a scaled version of Equation 2 subject to α that varies between 0 & 1 [3]. Given training vectors x_i where $i = 0, 1, ..., n$, $v \in (0, 1)$, where 0 denotes a lower limit of support vectors and 1 denotes an upper limit on errors made in training a model. Equation 4 shows the kernel function Q_{ij} of one class LibSVM i.e. a dot product of two training vectors.

Algorithm 2: One Class LibSVM Algorithm

Data: Training Dataset D_{tr}, Testing Dataset D_{te}
Result: List of class labels for test dataset C_{te}
Algorithm $OneClassLibSVM(D_{tr}, D_{te})$

1 $Class\_Label \leftarrow SVM.setTargetClass(D_{tr})$
2 $Model \leftarrow SVM.buildClassifier(D_{tr})$
3 Preprocessed dataset $T_p \leftarrow FilterTweets(D_{te})$
4 **for** ***each*** *instance* $i \in T_p$ **do**
5 Class $c \leftarrow Model.classifyTweets(i)$
6 $C_{te} \leftarrow c$
end
7 return C_{te}

Algorithm 2 describes basic modules of LibSVM that we implement in our classifier. We give an input of a training and testing dataset to the algorithm i.e. D_{tr} and D_{te} respectively. Training dataset contains a set of labeled feature vectors of only target class tweets. In steps 1 and 2, we set the target class label and build our model on training dataset. In step 3, we perform data pre-

[5] http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[6] http://www.cs.waikato.ac.nz/ml/weka/

processing on testing dataset and remove all garbage data. Steps 4 to 6 performs classification and predicts most likely class for a given instance of testing dataset.

$$min_\alpha\{\tfrac{1}{2}\alpha^T Q\alpha - e^T\alpha\}, \quad 0 \leq \alpha \leq C, \quad y^T\alpha = 0 \tag{2}$$

$$min_\alpha\{\tfrac{1}{2}\alpha^T Q\alpha\}, \quad 0 \leq \alpha \leq 1, \quad e^T\alpha = vn \tag{3}$$

$$Q_{ij} \equiv K(x_i, x_j) = (x_i.x_j) \tag{4}$$

We also implement leave-p-out cross validation strategy in both KNN and LibSVM classifiers. We perform a column-wise partition on both training and testing datasets and remove p feature/s at a time. We repeat this process for all features and run our proposed classifiers $2 * {}^mC_p$ times, where m is the size of feature space and p is the number of features we remove per iteration, $p = 1$ in our case. As a result of this, we get a $1*m$ matrix for one classifier, where each instance shows the overall accuracy of respective classifier.

5 Empirical Analysis and Performance Evaluation

5.1 Experimental Dataset

We conduct experiments on publicly available dataset so that our results can be replicated or used for benchmarking or comparison purposes. We download two datasets: UDI-TwitterCrawl-Aug2012[7] and ATM-TwitterCrawl-Aug2013[8]. UDI-TwitterCrawl-Aug2012 consists of 50 million tweets approximately and was collected in May 2011 [6]. ATM-TwitterCrawl-Aug2013 consists of 5 million English tweets and was collected in June 2011 [5]. We use language detection library[9] for Java for language identification (supports 53 languages) of tweets and find 29 different languages in the dataset. There are 85% English and 15% non-English tweets present in the experimental dataset. Initially we have $53,234,567$ tweets in our dataset. In this paper, we focus only on English language tweets, therefore we discard all non-English $(7,889,609)$ tweets and remain with a total of $45,344,958$ tweets. We perform a semi-supervised learning approach on experimental dataset and collect only hate & extremism promoting tweets. To avoid overfitting in classification we collect only $10,486$ labeled tweets for training dataset, which is a very small fraction of experimental dataset. We perform a random sampling on all 45.3 million tweets of experimental dataset and collect a random sample of 1 million tweets as testing (or validation) dataset. This dataset includes both hate promoting and unknown tweets.

[7] `https://wiki.engr.illinois.edu/display/forward/Dataset-UDI-TwitterCrawl-Aug2012`

[8] `https://wiki.engr.illinois.edu/display/forward/Dataset-ATM-TwitterCrawl-Aug2013`

[9] `https://code.google.com/p/language-detection/`

Table 6. Confusion Matrix And Accuracy Results

(a) KNN Classifier

		Predicted	
		Positive	**Unknown**
Actual	**Positive**	67,798	15,522
	Unknown	74,968	841,712

(b) LibSVM Classifier

		Predicted	
		Positive	**Unknown**
Actual	**Positive**	73,555	9,765
	Unknown	20,420	896,260

(c) Accuracy Results of Classifiers

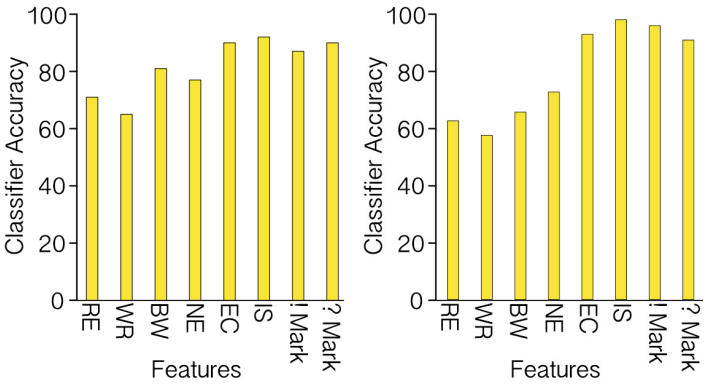
Classifier	**Precision**	**Recall**	**TNR**	**NPV**	**F-Score**	**Accuracy**
KNN	0.48	0.81	0.92	0.98	0.60	0.90
LibSVM	0.78	0.88	0.98	0.99	0.83	0.97

5.2 Experimental Results

To evaluate the performance of our proposed solution approach, we use basic measures of relevance used in information retrieval and machine learning. We asked 4 graduate students to manually annotate each tweet in the dataset and based upon their decisions we validate our results (we gave them simple instructions to annotate a tweet as posiitve if they find it hate and extremism promoting). We compute accuracy of our classifier in terms of precision, recall and f-score. Table 6(a) and 6(b) shows the standard confusion matrix for KNN and LibSVM classifiers. We execute our classifiers on a testing dataset of size 1 million records containing tweets from both target class (positive) and outliers. One class KNN algorithm classifies $142,766$ $(67,798 + 74,968)$ tweets as positive and $857,054$ $(15,522 + 841,712)$ tweets as unknown. Table 6(a) reveals that there is a misclassification of 18.6% and 8.2% in predicting target (positive) class and outlier (unknown) instances. Similarly, given an input of 1 million tweets, one class LibSVM algorithm predicts $103,975$ $(73,555 + 20,420)$ tweets as positive and $906,025$ $(9,765 + 896,260)$ tweets as unknown.

Table 6(b) shows that 11.7% and 2.2% of tweets are wrongly classified as positive and unknown respectively. Table 6(c) shows accuracy results (precision, recall, f-score) for both KNN and LibSVM classifiers. Table 6(c) reveals that overall LibSVM classifier (accuracy of 97%) outperforms than KNN classifier (accuracy of 90%). Results shows that precision, f-score and accuracy of LibSVM classifier are much higher in comparison to KNN classifier and similarly recall is reasonably high for LibSVM classifier.

We apply leave-p-out strategy for both KNN and LibSVM classifiers ($p = 1$) and compute their accuracy. As discussed in Section 3, we use 8 discriminatory features to classify a tweet as hate promoting or unknown. Figure 3 shows



(a) KNN (b) LibSVM

Fig. 3. Impact of Individual Feature on Overall Accuracy of A Classifier. RE= Religious, WR= War Related, BW= Bad Words, NE= Negative Emotions, EC= Emoticons, IS= Internet Slangs.

variance in overall accuracy of one class classifiers (KNN and LibSVM) after removing one feature vector at a time. Figure 3(a) reveals that if we remove religious or war related terms then the accuracy of KNN classifier decreases by 20 to 25%. Removing bad words or negative emoticons from feature vectors, accuracy falls down by 11 to 13%. Figure 3(a) reveals that internet slangs, emoticons and punctuations (! and ? marks) are less important features and doesn't affect the accuracy by a major difference but we can not neglect them completely because they affect the overall accuracy by 2 to 3%. Figure 3(b) reveals that in one class LibSVM classifier, presence of religious, war related terms, bad words and negative emotions plays an important role. And by removing any of these features, overall accuracy of classifier decreases by 20 to 45%. Ignoring presence of internet slangs and exclamation marks doesn't affect accuracy. Unlike KNN classifier, removing emoticons and question marks decreases the performance by a reasonable rate. The reason of this misclassification is the presence of noisy content and sparsity in datasets. Feature space of testing dataset is a matrix of size 1M*8, where 70% of entries are 0.

6 Conclusion

Hate and extremism promoting users and Tweets are prevalent on Twitter. We observe presence of tweets containing hashtags indicating hate and extremism and also tweets which do not contain such hashtags but are hate and extremism promoting. We conduct a manual analysis of tweets and identify linguistic features which can be used as discriminators for the task of identifying hate and extremism promoting tweets. We demonstrate a correlation between such tweets and features like presence of war, religious, negative emotions and offensive terms. We train a one-class SVM and KNN on 10,486 positive class tweets

and observe an F-Score of 0.83 and 0.60 respectively. We implement a leave one out strategy and examine the influence of each discriminatory feature on overall accuracy of classifiers. Based upon the accuracy results, we conclude that presence of religious, war related terms, offensive words and negative emotions are strong indicators of a tweet to be hate promoting. Unlike KNN classifier, presence of internet slangs and question mark plays an important role in LibSVM classifier.

References

1. Agrawal, S., Sureka, A.: Learning to classify hate and extremism promoting tweets. JISIC (2014)
2. Berger, J., Strathearn, B.: Who matters online: Measuring influence, evaluating content and countering violent extremism in online social networks. The international centre for the study of radicalization and political violence (2013)
3. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011)
4. Kwok, I., Wang, Y.: Locate the hate: Detecting tweets against blacks. In: Twenty-Seventh AAAI Conference on Artificial Intelligence (2013)
5. Li, R., Wang, S., Chang, K.C.C.: Towards social data platform: automatic topic-focused monitor for twitter stream. Proceedings of the VLDB Endowment 6(14), 1966–1977 (2013)
6. Li, R., Wang, S., Deng, H., Wang, R., Chang, K.C.C.: Towards social user profiling: unified and discriminative influence model for inferring home locations. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1023–1031. ACM (2012)
7. Liebrecht, C., Kunneman, F., van den Bosch, A.: The perfect solution for detecting sarcasm in tweets# not. Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (2013)
8. Martinez-Romo, J., Araujo, L.: Detecting malicious tweets in trending topics using a statistical analysis of language. Expert Systems with Applications 40(8), 2992–3000 (2013)
9. O'Callaghan, D., Greene, D., Conway, M., Carthy, J., Cunningham, P.: Uncovering the wider structure of extreme right communities spanning popular online networks. In: Web Science Conference, pp. 276–285 (2013)
10. Reyes, A., Rosso, P., Buscaldi, D.: From humor recognition to irony detection: The figurative language of social media. Data & Knowledge Engineering 74, 1–12 (2012)
11. Wadhwa, P., Bhatia, M.P.S.: Tracking on-line radicalization using investigative data mining. In: NCC, pp. 1–5 (2013)
12. Xiang, G., Fan, B., Wang, L., Hong, J., Rose, C.: Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 1980–1984. ACM (2012)
13. Yang, M.C., Lee, J.T., Lee, S.W., Rim, H.C.: Finding interesting posts in twitter based on retweet graph analysis. In: SIGIR, pp. 1073–1074 (2012)