Scam Detection in Twitter

Xiaoling Chen R. Chandramouli K. P. Subbalakshmi
Department of Electrical and Computer Engineering
Stevens Institute of Technology

Abstract

Twitter is one among the fastest growing social networking services. This growth has led to an increase in Twitter scams (e.g., intentional deception). There is relatively little effort in identifying scams in Twitter. In this paper, we propose a semi-supervised Twitter scam detector based on a small labeled data. The scam detector combines self-learning and clustering analysis. A suffix tree data structure is used. Model building based on Akaike and Bayes Information Criteria is investigated and combined with the classification step. Our experiments show that 87% accuracy is achievable with only 9 labeled samples and 4000 unlabeled samples, among other results.

Keywords :Social network security, Twitter scam, suffix tree, clustering analysis, semi-supervised learning, AIC, BIC

1 Introduction

In recent years, social networking sites, such as Twitter, LinkedIn and Facebook, have gained notability and popularity worldwide. Twitter as a microblogging site, allows users to share messages and discuss using short texts (no more than 140 characters), called tweets. The goal of Twitter is to allow users to connect with other users (followers, friends, etc.) through the exchange of tweets.

Spam (e.g. unwanted messages promoting a product) is an ever-growing concern for social networking systems. The growing popularity of Twitter has sparked a corresponding rise in spam tweets. Twitter spam detection has been getting a lot of attention. There are two ways in which a user can report spams to Twitter. First, a user can click the "report as spam" link on their Twitter homepage. Second, a user can simply post a tweet in the format of "@spam@username" where @username is the spam account. Also, different detection methods (see, Section 3) have been proposed to detect spam accounts in Twitter. However, Twitter scam detection has not received the same level of attention. Therefore, methods to successfully detect Twitter scams are important to improve the quality of service and trust in Twitter.

A primary goal of Twitter scams is to deceive users then lead them to access a malicious website, believe a false message to be true, etc. Detection of Twitter scams is different from email scam detection in two aspects. First of all, the length (number of words or characters) of a tweet is significantly shorter than an average email length. Therefore some of the features indicating email scam are not good indicators of Twitter scams. For example, the feature "number of links" indicating the number of links in an email is used in email phishing detection. However, due to the 140 character limit usually there is at most one link in tweets. Further, Twitter offers URL shortening services and applications and the shortened URLs can easily hide malicious URL sources. Thus most of the features about URL links in the email context are not applicable for tweet analysis. Second, the constructs of emails and tweets are different. In Twitter, a username can be referred in @username format in the tweet. A reply message is in format @username+message where @username is the receiver. Also, a user can use the hashtag "#" to describe or name the topic in a tweet. Therefore, due to a tweet's short length and the special syntax a predefined, fixed set of features will not be effective to detect scam tweets.

In this paper, we propose and evaluate a semi-supervised tweet scam detection method that com-

bines self-learning and clustering analysis. We use a detector based on the suffix tree data structure [3] as the basic classifier for semi-supervised learning. Differing from other techniques, the suffix tree approach can compare substrings of an arbitrary length. The substring comparison has particular benefits in Twitter scam detection. For example, since the writing style in Twitter is typically informal we frequently observe many typographical errors. Two words like "make money" may appear as "makemoney". If we consider each word as a unit then "makemoney" will be treated as a new word and cannot be recognized. Instead, if we treat the words as character strings, then we will be able to recognize this substring.

This paper is organized as follows. In Section 2 the Twitter scam problem is introduced. The related work is discussed in Section 3. In Section 4, the suffix tree algorithm is described and the proposed small sample learning method based on suffix tree is presented in Section 5. Section 6 introduces the data collected and used for experimental analysis. Experiment analysis and results are presented in Section 7. Section 8 contains the main conclusions.

2 Scams in Twitter

Twitter has been a target for scammers. Different types of scams use different strategies to misguide or deceive Twitter users. The techniques and categories of scams keep evolving constantly. Some Twitter scams can been categorized as follows (e.g., [8]): (1) straight cons; (2) Twitomercials or commercial scam and (3) phishing and virus spreading scams.

2.1 Straight cons: Straight cons are attempts to deceive people for money. For example, the "Easy-money, work-from-home" schemes, "Promises of thousands of instant followers" schemes and "money-making with Twitter" scams fall in this category [9].

In an "easy-money work-from-home" scammers send tweets to deceive users into thinking that they can make money from home by promoting products of a particular company. But, in order to participate in the work from home scheme users are asked to buy a software kit from the scammer, which will turn out to be useless. Another strategy that is used by scammers is to post a link in the tweet that points to fraudulent website. When one sign-ups in that website to work from home, users are charged a small fee initially. However, if the user pays using a credit card, the credit card will be charged for a recurring monthly membership fee and it is almost impossible to get the money back.

In a typical "promises of thousands of instant followers" scam, the scammers claim that they can identify thousands of Twitter users who will automatically follow anyone who follow them. Twitter users will be charged for this service. But, the users' account typically ends up in a spammer list and banned from Twitter.

In a "money-making with Twitter" scam, scammers offer to help users to make money on Google or Twitter. When someone falls for this scam, they are actually signing up for some other service and is charged a fee. Another example is when one may get a tweet apparently from a friend asking to wire cash since she is in trouble. This happens when a scammer hijacks the friend's Twitter account and pretends to be the friend.

Several examples of Twitter scams in this category include the following:

Single Mom Discovers Simple System For Making Quick And Easy Money Online with Work-At-Home Opportunities! http://tinyurl.com/yc4kadd

#NEWFOLLOWER Instant Follow TO GET 100 FREE MORE TWITTER FOLLOWERS! #FOLLOW http://tinyurl.com/255lgwg

Visit my online money making website for tips and guides on how to make money online. http://miniurls.it/beuKFV

2.2 Twitomercial: Commercial spam is an endless repetitive stream of tweets by a legitimate business while a commercial scam or Twitomercial consists of tricks employed by businesses with a malicious intent. The teeth whitening scam is a typical example of a commercial scam. Here, the tweet claims that one can get a free trial teeth whitening package and a HTTP link to their fake website is included. In the fake website one is instructed to

sign up for the free trial and asked to pay only the shipping fee. But, in fact, a naive user will also be charged a mysterious fee and also will receive nothing for the payment. An example of the teeth whitening scam is the following tweet:

Alta White Teeth Whitening Pen - FREE TRIAL Make your teeth absolutely White. The best part is It is free! http://miniurls.it/cyuGt7

2.3 Phishing and virus spreading scams: Phishing is a technique used to fool people into disclosing personal confidential information such as the social security number, passwords, etc. Usually the scammers masquerade as one's friend and send them a message that includes a link to a fake Twitter login page. The message will be something like "just for fun" or "LOL that you?". Once the user enters their login and password in the fake page that information will be used for spreading Twitter spam or virus. The format of the virus spreading scam is almost the same as that of the phishing scam. Therefore we group them into the same category. Different from phishing, virus spreading scam includes a link which will upload malware onto the computer when it is clicked. An example of the phishing tweet is shown below:

Hey, i found a website with your pic on it LOL check it out here twitterblog . access-logins . com / login

---

3 Related work

Twitter spam detection has been studied recently. The existing work mainly focuses on spammer detection. In [10], the behavior of a small group of spammers was studied. In [4], the authors proposed a naive Bayesian classifier to detect spammer Twitter accounts. They showed that their detection system can detect spammer accounts with 89% accuracy. In [7], the authors collected a large data. 39 content attributes and 23 user behavior attributes were defined and a SVM classifier was used to detect a spammer's Twitter account. In [5], a honeypot-based approach for uncovering social spammers in online social systems including Twitter and MySpace was proposed. In [6], the authors studied and compared five different graph centrality algorithms to detect Twitter spammer accounts.

3.1 Suffix tree (ST) based classification: The suffix tree is a well studied data structure which allows for fast implementation of many important string operations. It has been used to classify sequential data in many fields including text classification. In [3], a suffix tree approach was proposed to filter spam emails. Their results on several different text corpora show that character level representation of emails using a suffix tree outperforms other methods such as a naive Bayes classifier. In this paper, we use the suffix tree algorithm proposed in [3] as a basic method to classify tweets.

3.2 Semi-supervised methods: Supervised techniques have been used in text classification application widely[1]. Usually it requires a large number of labeled data to train the classifiers. Assigning class labels for a large number of text documents requires a lot of effort. Therefore we investigate semi-supervised techniques. In [2], the authors presented a theoretical argument showing that unlabeled data contain useful information about the target function under common assumptions.

To the best of our knowledge, the semi-supervised learning has not been considered for the Twitter scam detection. In this paper, we propose a semi-supervised learning method combining model-based clustering analysis with the suffix tree detection algorithm to detect Twitter scam.

4 Suffix tree algorithm

4.1 Scam detection using Suffix tree: The suffix tree algorithm we use here is a supervised classification method and can be used to classify documents [3]. In the scam detection problem, given a target tweet d, and suffix trees T_S and T_{NS} for the two classes, we can solve the following optimization problem to find the class of the target tweet:

$$G = \arg \max_{\theta \in \{S, NS\}} score(d, T_\theta) \tag{4.1}$$

The models T_S and T_{NS} are built using two training data sets containing scam and non-scam tweets, respectively. In Twitter scam detection, the false positive errors are far more harmful than the false negative ones. Misclassification of non-scam tweets will upset the users and may even result in some sort of an automatic punishment to the user. To implement (4.1) we compare ratio between scam score and non-scam score with a threshold to determine the scam or not scam. The threshold can be computed based on the desired false positive rate or false negative rate. Figure 1 shows the flowchart of the suffix tree score based scam detection algorithm.

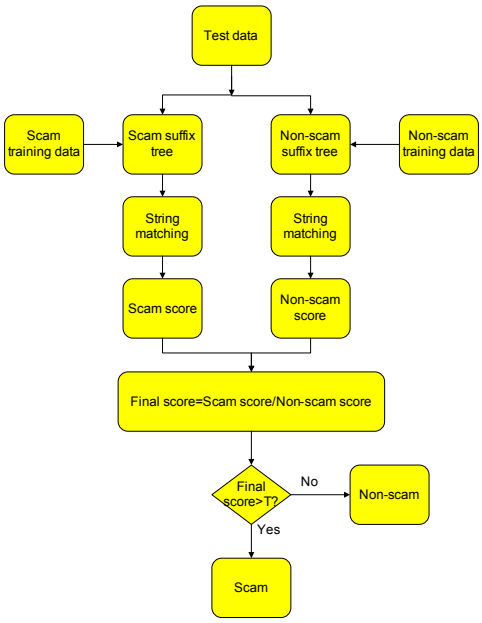

Figure 1: Flowchart of suffix tree based representation and scam detection algorithm

4.2 Suffix tree construction: The suffix tree structure used here is different from the traditional suffix tree in two aspects: label each node but not edges; no terminal character. Labeling each node makes the frequency calculation more convenient and the terminal character does not play any role in the algorithm and therefore omitted. To construct a suffix tree from the string, first the depth of the tree is defined. Then the suffixes of the string are defined and inserted into the tree. A new child node will only be created if none of the existing child nodes represents the character we consider. Algorithm 1 gives the suffix tree construction scheme we use.

Algorithm 1: The suffix tree building algorithm

(1)Define tree length N.
(2)Create suffixes w(1)-w(n), $n = \min\{N, length(s)\}$.
(3)For i=1 to n, $w(i) = m_1 m_2 \cdots m_j$, j=length(w(i)):
From the root,
For $k = 1$ to j:
If m_k in level k:
increase the frequency of node m_k by 1,
else:
create a node for m_k, frequency=1
move down the tree to the node of m_k

Let us consider a simple example for illustration. Suppose we want to build a suffix tree based on the word "seed" with tree depth $N = 4$. The suffixes of the string are w(1)= "seed", w(2)= "eed", w(3)= "ed" and w(4)= "d". We begin at the root and create nodes for $w(1)$ and $w(2)$. When we reach $w(3)$, "e" node already exists in level 1 and we just increase its frequency by 1. Then a "d" node is created in level 2 after the "e" node. Figure 2 shows the suffix tree built based on "deed" and "seed". "d(2)" means the node represents the character "d" and its frequency is 2. For more details about the characteristics of the suffix tree, we refer to [3].

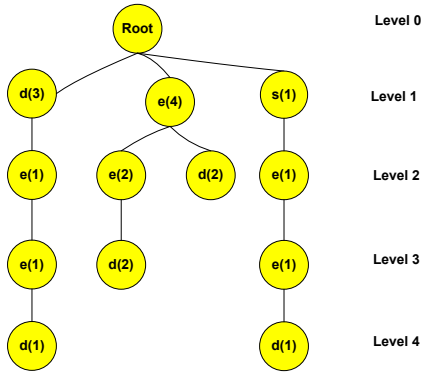

Figure 2: Suffix tree for "deed" and "seed"

4.3 Scoring scam: Given a target tweet d and a class tree T, d can be treated as a set of substrings. The final score of the tweet is the sum of the individual scores each substring gets as shown in (4.2).

$$score(d, T) = \sum_{i=0}^{M} match(d(i), T) \tag{4.2}$$

$match(d(i), T)$ calculates the match between each substring and class tree T using (4.3). Suppose $d(i) = m_1 \cdots m_k$, where m_j represents one character, the match score of $d(i)$ is the sum of the significance of each character in the tree T. The significance is computed using a significance function $\phi()$ on the conditional probability p of each character m_k. The conditional probability can be estimated as the ratio between the frequency of m and the sum of the frequencies of all the children of m's parent as given in (4.4). n_m is the set of all child nodes from m's parent.

$$match(d(i), T) = \sum_{j=0}^{k} \phi[p(m_j)] \tag{4.3}$$

$$\phi(p(m)) = \frac{1}{1 + exp(-p(m))}, \quad p(m) = \frac{f(m)}{\sum_{L \in n_m} f(L)} \tag{4.4}$$

5 Semi-supervised Learning for Scam Detection

Self-training is a commonly used semi-supervised learning method [11]. Since self-training uses the unlabeled data which are predicted by itself, the mistake in the model will enforce itself and it is vulnerable to the training bias problem. Three aspects play important role in improving the performance of self-training. First, choose a classifier with good performance. Second, obtain informative labeled data before training. Third, set a confidence threshold to pick the high confident unlabeled data into training set in each iteration.

In this paper, we use the suffix tree based classifier as described previously, for two reasons. First, the ability of a suffix tree to compare any length of substrings is useful for Twitter data analysis. Second, suffix trees can be updated very efficiently as new tweets are collected.

To obtain a set of informative labeled data, we propose a model-based clustering analysis. Different types of Twitter scams have different formats and structures as discussed in Section 2. To make the detector more robust the labeled training set should cover a diverse set of examples. However, in Twitter, scammers often construct different scams using minor alterations from a given tweet template. In this case, if we randomly pick samples to label for the training set, especially with a small number of samples, there is a high possibility that the training set will not be diverse and may be unbalanced. Unbalanced means that we may pick several samples for the same scam type while missing the samples of some other scam type. To address this problem, clustering analysis before training will provide useful information to select the representative tweets for labeling. In this paper, we use the K-means clustering algorithm to cluster the training data. Euclidean distance is used to compute the distance metric. To select the most informative samples in the training data the number of clusters should also be considered. Here we adopt two model selection criteria: Akaike information criterion (AIC) and the Bayesian information criterion (BIC). After the best models are selected based on AIC and BIC one sample which is closest to the centroid in each cluster will be selected to be labeled and used as the initial training data.

5.1 LSA feature reduction: For most document clustering problems the vector space model (VSM) is a popular way to represent the document. In this paper, we first pre-process the tweet using three filters: a) remove all punctuations; b) remove all stop-words; and c) stem all remaining words. The stop-words we used is from the Natural Language Toolkit stopwords corpus [13], which contain 128 English stop-words. Then each tweet is represented as a feature vector. Each feature is associated with a word occurring in the tweet. The value of each feature is the normalized frequency of each word in the tweet. Since each tweet is can be up to 140 characters, the feature number m is large and feature space has a high dimension. Thus clustering for documents is very poor in terms of scalability and is time consuming. Therefore we use Latent Semantic Analysis (LSA) to reduce the feature space. LSA decomposes a large term-by-document matrix into a set of orthogonal factors using singular value decomposition (SVD). The LSA can reduce the dimension in the feature space and still provide a robust space for clustering. Since different types of scam may contain some cer-

tain keywords, the clustering procedure will cluster the similar scam tweets into the same cluster and the pre-process step will not affect the clustering result.

5.2 Model-based clustering analysis: To select the most informative samples from the data, first we need to cluster the data and select the ones which can best represent the whole data set. We use model-based clustering approach, where each cluster is modeled using a probability distribution and the clustering problem is to identify these distributions.

Each tweet is represented as a vector containing a fixed number of attribute values. Given tweet data $x^n = (x_1, \cdots, x_n)$ each observation has p attributes $x_i = (x_{i0}, \cdots, x_{ip})$. Let $f_k(x_i|\theta_k)$ denote the probability density of x_i in the kth group, where θ_k is a parameter(s) in the kth group, with total number of groups equal to G. Usually, the mixture likelihood [12] of the model can be written as (5.5) where γ_i is the cluster label value, i.e., $\gamma_i \in \{1, 2, \cdots, G\}$. For example, $\gamma_i = k$ means that x_i belongs to the kth cluster:

$$L(\theta|x^n) = \prod_{i=1}^{n} f_{\gamma_i}(x_i|\theta_{\gamma_i}) \tag{5.5}$$

In our study, we assume $f_k(x_i|\theta_k)$ to be a multivariate Gaussian model. Then $\theta_k = (u_k, \Sigma_k)$ where u_k is the mean vector of the k cluster and Σ_k is the covariance matrix. We use the hard assignment K-means clustering to cluster the data. Clusters are identical spheres with centers u_k and associated covariance matrices $\Sigma_k = \lambda I$. Then

$$f_k(x_i|u_k, \Sigma_k) = \frac{\exp\{-\frac{1}{2}(x_i - u_k)^T(x_i - u_k)\}}{(2\pi)^{p/2}\lambda^{(p+2)/2}}$$

Then the log likelihood equation (5.5) becomes

$$ln(L(\theta|x^n)) = ln(\prod_{i=1}^{n} \frac{1}{(2\pi)^{\frac{p}{2}}\lambda^{\frac{p+2}{2}}}) + \sum_{i=1}^{n} -\frac{1}{2}(x_i - u_{\gamma_i})^T(x_i - u_{\gamma_i})$$

Since $ln(\prod_{i=1}^{n} \frac{1}{(2\pi)^{p/2}\lambda^{(p+2)/2}})$ depends on the data and independent of the model used it is a constant if the data is not changed. Then we can omit this in the log likelihood function. Then

$$ln(L) = -\frac{1}{2}\sum_{i}^{n}(x_i - u_{\gamma_i})^T(x_i - u_{\gamma_i}) = -\frac{1}{2}\sum_{j=1}^{G} R_{ssj} \tag{5.6}$$

Where R_{ssj} is residual sum of squares in jth cluster.

The next question we address is how to determine G. The model selection process is to select an optimum model in terms of low distortion and low complexity. We adopt two popular selection criteria, Akaike Information Criterion (AIC) and Bayesian information criterion (BIC) for optimal model selection. In our problem, the information criterion becomes

$$AIC = \frac{1}{2}\sum_{j=1}^{G} R_{ssj} + pG$$

$$BIC = \frac{1}{2}\sum_{j=1}^{G} R_{ssj} + \frac{pG}{2}\ln(n)$$

By associating the data with a probability model, the best fitting model selected by AIC or BIC is the one assigns the maximum penalized likelihood to the data.

5.3 Twitter Scam Detection: To avoid the bias of self-training, a confidence number is used to include the unlabeled data into training set in each iteration. In each prediction, a scam score h_{scam} and a non-scam score h_{nscam} is obtained for each unlabeled tweet. Here we define the ratio $hr = h_{scam}/h_{nscam}$ as the selection parameter. The higher the hr is, the more confidence that the tweet is scam. Then in each iteration, the C scam and non-scam tweets with the highest confidence are added to the training set. The steps of the proposed semi-supervised learning method is given in Algorithm 2.

We point out that the confidence number C and the suffix tree depth have to be chosen in our algorithm. In the experimental analysis section we describe how these numbers affect the performance of the detector.

6 Twitter Data Collection

In order to evaluate the proposed scam detection method we use a collection of Tweets that include scams and legitimate data. To the best of our knowledge there is no such collection available publicly prior to this work. Therefore we developed a crawler to collect Tweets using the API methods provided by Twitter. It is impossible to collect all the tweets exhaustively. Also, labeling all the

Algorithm 2: Semi-supervised learning based on clustering analysis and suffix tree

Input: U: a set of unlabeled tweets;
F: suffix tree algorithm;
C: confidence number;
K: maximum cluster number;
(1)Preprocess U, feature matrix D;
(2)Feature reduction by LSA, reduced feature matrix DD;
(3)Clustering DD into N clusters $c_1, \cdots, c_N$ $N \in (2, \cdots, K)$ using K-means and compute AIC or BIC;
(4)Select the model with minimum AIC or BIC;
(5)Select one sample in each cluster and label, as L;
(6)Update $U = U - L$;
(7)**while** U is not empty:
update F with L;
predict U using F, return hr;
sort the tweet according to the hr in descend;
select C tweets from the front of the sorted list, add to L;
select C tweets from the end of the sorted list, add to L;
update U, L;
Return F;

collected tweets manually is very difficult. For these reasons we set a limit on the number of tweets in our data corpus.

As a first approximation to collect scam tweets we queried Twitter using frequent English stop words, such as "a","and","to","in", etc. To include a significant number of scam tweets into our data corpus, we queried Twitter using keywords such as "work at home", "teeth whitening", "make money" and "followers". Clearly, the queries could return both scams as well as legitimate tweets. We collected tweets from May 15 to May 20, 2010. Totally, we collected about 12000 tweets. Twitter scammers usually post duplicate or highly similar tweets by following different users. For instance, the scammer may only change the HTTP link in the tweet while the text remains the same.

After deleting duplicate and highly similar tweets 9296 unique tweets were included in our data set. Then we divided our data set into two subsets, namely, training dataset and test dataset. We randomly picked 40% of the tweets as the test data. Thus the training data set contained 5578 tweets and the test data set contained 3718 tweets. By using the semi-supervised method we only needed to label a small number of tweets in the training data set. However, in order to evaluate the performance of the detector the test data set needed to be labeled as well. In order to minimize the impact of human error three researchers worked independently to label each tweet. They were aware of the popular Twitter scams and labeled a tweet as non-scam if they were not confident if the tweet was a scam. The final labeling of each tweet was based on the majority voting considering the labeling of the three researchers. After labeling we obtained 1484 scam tweets and 2234 non-scam tweets in the test set. For the training data set only a small number of tweets were labeled.

7 Experimental Results

In this section, we present the results of experiments carried on our collected data set to investigate the effectiveness of the proposed semi-supervised scam detector. First the evaluation metrics used in this paper are discussed and followed by a discussion of the obtained results.

7.1 Evaluation metrics: Table 1 shows the confusion matrix for the scam detection problem.

Table 1: A confusion matrix for scam detection

		Predicted	
		Scam	Non-scam
Actual	Scam	A(+ve)	B(-ve)
	Non-scam	C(-ve)	D(+ve)

In Table 1, A is the number of scam tweets that are classified correctly. B represents the number of scam tweets that are falsely classified as non-scam. C is the number of non-scam tweets that are falsely classified as scam while D is the number of non-scam tweets that are classified correctly. The evaluation metrics we use are:

- **Accuracy** is the percentage of tweets that are classified correctly, $Accuracy = \frac{A+D}{A+B+C+D}$.
- **Detection rate (R)** is the percentage of scam tweets that are classified correctly, $R = \frac{A}{A+B}$.
- **False positive(FP)** is the percentage of non-scam tweets that are classified as scam, $False\ positive = \frac{C}{C+D}$.
- **Precision (P)** is the percentage of predicted scam tweets that are actually scam. It is defined as $P = \frac{A}{A+C}$.

7.2 Experiment results and analysis: We begin by comparing the Suffix tree algorithm with the Naive Bayesian (NB) classifier on a small amount of labeled data set. First we randomly picked 200 tweets from the training set of which 141

were not scam and 51 were scams. We then built the ST classifier and NB classifier on a training set with N samples ($N/2$ are scam samples and $N/2$ are non-scam samples) respectively. Then the classifiers were tested on the same test data set. The depth of ST was set to 4 in this experiment. The N samples were randomly picked from the 200 labeled tweets and this procedure was repeated 10 times to compare the performance of Suffix tree and Naive Bayesian. For Naive Bayesian, punctuation and stop-words were first removed from the tweets and stemming was implemented to reduce the dimension of features. For both methods, the threshold was changed from 0.9 to 1.3 in increments of 0.01 and the threshold which produced the highest accuracy was chosen. Table 2 shows the average detection results of Naive Bayesian classifier and Suffix tree for different values of N.

Table 2: Results of supervised methods on small training data

N	Method	Accuracy	R	FP	P
N=10	NB	62.42%	87.65%	54.30%	56.26%
	ST	65.87%	78.40%	42.42%	57.43%
N=30	NB	68.95%	95.90%	48.93%	57.45%
	ST	74.10%	78.32%	28.67%	64.62%
N=50	NB	72.57%	94.25%	41.78%	60.54%
	ST	74.65%	79.23%	28.36%	65.16%
N=100	NB	72.21%	97.13%	44.30%	59.37%
	ST	77.63%	79.18%	23.38%	69.03%

From Table 2, we can see that Suffix tree outperforms Naive Bayesian with lower false positive rate and higher detection accuracy. As expected, increasing N improves the performance of both methods. Using only 10 samples as training data we can correctly classify about 65% of the tweets in test data using Suffix tree. While using 100 samples we can achieve about 78% accuracy. Although 65% and 78% may not be as high as desired, nevertheless this experiment sheds light on the feasibility of the self-learning detector. An unexpected result is that Naive Bayes classifier achieves very high detection rate R in all the cases. A possible explanation is that after the preprocessing steps the feature words in the scam model are less diverse than the features words in the non-scam model. This is because scam tweets usually contain a HTTP link and more punctuation. In the test step, when a word does not occur in the training data previously, a smoothing probability will be assigned to it. Since the number of features in scam model is smaller than in the non-scam model, the smoothing probability will be higher in the scam model thus resulting in a higher final score. Then the NB will classify most of the tweets in the test data as scam. This results in the high detection and high false positive rates.

We then evaluated the self-learning methods on the data set. We implemented the K-means algorithm to cluster the training data set and selected one sample from each cluster to be labeled. The feature matrix is reduced to a lower dimension by LSA with $p = 100$. To compute the AIC and BIC the cluster number N was changed from 2 to 40. For each N, 10 runs were used and the maximum value of $ln(L)$ in (5.6) was used for the model to compute the AIC and BIC values. For AIC, $N = 9$ resulted in the best model while for BIC, $N = 4$ was the optimal value. Since BIC includes a higher penalty the optimum value of N using BIC is smaller than that of AIC. We changed p to some other numbers and similar results were achieved thus we used $p = 100$ in the following experiments.

We also randomly selected 9 samples to label in order to evaluate the effectiveness of the clustering step. In this experiment, the tree depth was set to 4 and in each iteration, $C = 200$ scam samples that were decided with the (rank ordered) highest confidence levels and similarly chosen non-scam samples were added to L to update the suffix tree model. Figure 3 shows the receiver operating characteristic (ROC) curve of the different methods. From this figure, we can see that the unlabeled data are useful in Twitter scam detection when proper semi-supervised learning is used. The proposed method can detect about 81% of the scams with low false positive (8%) rates using 9 labeled samples and 4000 unlabeled samples.

Figure 4 shows the detection accuracies after each iteration with and without clustering. The performances of AIC and BIC are similar while AIC achieves a slightly better result. We notice that clustering results in a higher accuracy in the

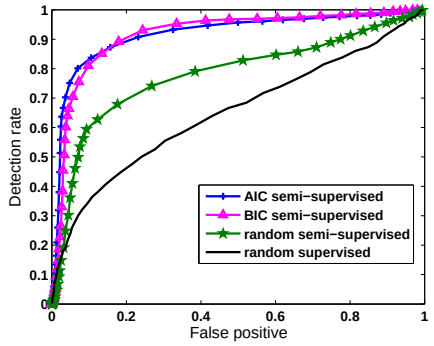

Figure 3: ROC curve of different methods

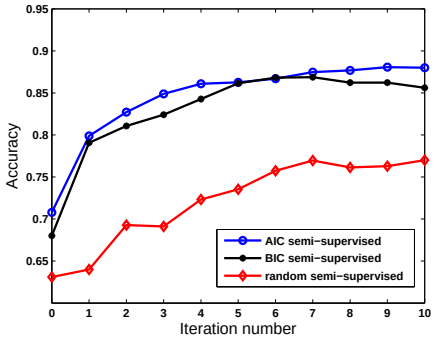

Figure 4: Accuracy of each iteration in self-learning

0th iteration compared to randomly selection. This also results in higher accuracies in the following iterations since the error in the model propagates. Therefore this indicates the importance of labeled data selection as addressed in this paper. Since AIC achieves the best result it was adopted in the following experiments.

To build trees as deep as a tweet is long is too computationally expensive. Moreover, the performance gain from increasing the tree depth may be negative. Therefore, we examine the tree depth to be 2, 4 and 6. We found that when the depth is set to 2 and $C = 200$, after 10 iterations about 72% accuracy was achieved. About 87% accuracy was achieved when the depths were 4 and 6. Since depth 6 does not outperform depth 4 but increases the tree size and the computational complexity, we choose depth to be 4 in the following experiments.

We then changed the value of C in each iteration to see how it influences the detection results. In this experiment the 9 samples selected by AIC was used to train the suffix tree initially. We changed C to be 50, 100, 200, 300, 400 respectively and for each C a total of 4000 unlabeled samples were used in the training process. Figure 5 shows the detection results. It is seen that when $C = 200$ the proposed method achieves a best accuracy rate of 87%. Increasing the value of C may decrease the performance since it will introduce errors into the training model. Thus picking a suitable value of C is important. In the following experiment, C was set to be 200.

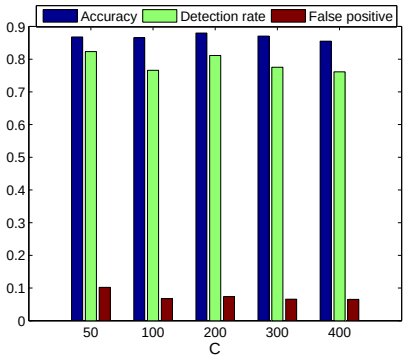

Figure 5: Results of semi-supervised method on different C

Recall that N is the number of labeled training data. Using AIC and BIC to choose N results in a small value for it. We may ask: if we use a larger labeled training set, can we achieve better results? To investigate this we considered four possible value, N=9, 50, 200, and 400. Different values of N were set in the K-means algorithm for clustering and one sample in each cluster was selected to label. Since we observed that $C = 200$ and depth 4 resulted in the best result, we compared different value of N under this set up and over 10 iterations. Thus a total of 4000 unlabeled data were used in the training process. Figure 6 shows the accuracies at each iteration with different values for N. From the result, we can see that, using more labeled training data, the initial classifier achieves higher accuracy. But after 10 iterations, the difference is not significant. The accuracy values are between 87-89%. When N=400, we achieve about 88% accuracy and for $N = 9$ determined using AIC we achieve about 87%. This result also illustrates

the advantage of the proposed clustering method before training. When $N = 9$, the initial classifier can only achieve an accuracy of 70.39%. However, after self-training 4000 unlabeled data, we observe that the results are competitive to the case with a larger value of N. We can explain this as follows. Since the labeled data samples are selected to be representative of the entire training set, it has a higher capability to correctly train the unlabeled data.

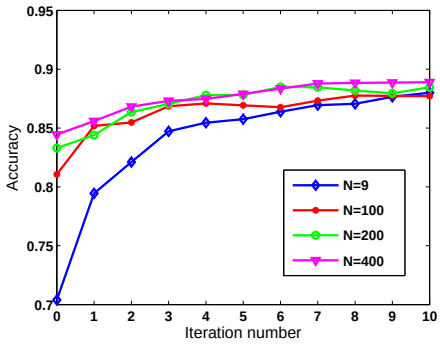

Figure 6: Results of semi-supervised method on different N

Consider a much larger tweet collection, the optimal number of clusters is expected to be larger. The clustering procedure will be more computational complexity since AIC or BIC should be calculated on different N. Thus more advanced methods to find the optimum clustering model is desired. An easy alternative is to select a reasonable N instead of using AIC or BIC in practical. Also, the tree size is expected to be larger when consider a larger corpus. However, since new nodes will be created only if the substrings have not been encountered previously, if the alphabet and the tree depth are fixed, the size of the tree will increase with a decreasing rate.

8 Conclusion

In this paper, we addressed the problem of Twitter scam detection using a small amount of labeled samples. Experiment results show that Suffix Tree outperforms Naive Bayesian for small training data and the proposed method can achieve 87% accuracy when using only 9 labeled tweets and 4000 unlabeled tweets. For some cases, unexpectedly, Naive Bayes classifier achieves high detection rates.

References

[1] J. M. Xu, G. Fumera, F. Roli and Z. Hu. Zhou, *Training SpamAssassin with Active Semi-supervised Learning*, CEAS 2009 - Sixth Conf. on Email and Anti-Spam July 16-17, 2009, Mountain View, California USA.

[2] K. Nigam, A. McCallum and T. M. Mitchell, *Semi-Supervised Text Classification Using EM*, In Chapelle, O., Zien, A., and Scholkopf, B. (Eds.) Semi-Supervised Learning. MIT Press: Boston. 2006.

[3] R. Pampapathi, B. Mirkin and M. Levene, *A Suffix Tree Approach to Anti-Spam Email Filtering*, Machine Learning, Kluwer Academic Publishers, 2006.

[4] A. H. Wang, *Don't Follow Me: Spam Detection in Twitter*, Int'l Conf. on Security and Cryptography (SECRYPT), 2010.

[5] K. Lee, J. Caverlee and S. Webb, *Uncovering social Spammers: Social Honeypots+Machine learning*, SIGIR'10, July19-23,2010, Geneva, Switzerland.

[6] D. Gayo-Avello and D. J. Brenes, *Overcoming Spammers in Twitter-A Tale of Five Algorithms*, CERI 2010, Madrid, Espana, pp. 41–52.

[7] F. Benevenuto, G. Magno, T. Rodrigues and V. Almeida *Detecting Spammers on Twitter*, CEAS 2010-Seventh annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conf., July 13-14, 2010, Redmond, Washington, US.

[8] *Twitter Spam: 3 Ways Scammers are Filling Twitter With Junk*, http://mashable.com/2009/06/15/twitter-scams/, 2009.

[9] *Twitter Scam Incidents Growing: The 5 Most Common Types of Twitter Scams – and 10 Ways to Avoid Them*, http://www.scambusters.org/twitterscam.html,2010.

[10] S. Yardi, D. Romero, G. Schoenebeck and D. Boyd, *Detecting spam in a Twitter network*, First Monday, 15(1), 2010.

[11] X. Zhu, *Semi-Supervised Learning Literature Survey*, Computer Sciences Technical Report 1530, Univ. of Wisconsin, Madison, 2006.

[12] C. Fraley and A. E. Raftery, *How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis*, The Computer Journal, 41, pp. 578-588, 1998.

[13] *Natural Language Toolkit*, http://www.nltk.org/Home, 2010.